

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Gestão Integrada de Operações de Recolha e Transporte de Biomassa

Francisco Areias Ribeiro Ferreira Capa

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Doutor Jorge Pinho de Sousa

Co-Orientador: Professora Doutora Alexandra Sofia da Fonseca Marques

12 de Julho de 2017

Resumo

A dissertação apresentada aborda um problema de sincronização das operações de recolha e transporte de biomassa florestal. A dissertação foi desenvolvida na instituição INESC TEC no âmbito do projeto de investigação e desenvolvimento (I&D) *EasyFlow* financiado pela Fundação para a Ciência e Tecnologia (FCT).

A biomassa tem vindo a ser classificada como uma das principais fontes de energia para a obtenção de uma futura segurança energética a nível mundial. A disponibilidade temporária e geograficamente fragmentada dos resíduos florestais tornam a otimização de todo o processo de recolha e transporte muito importante para a redução de custos de uma empresa.

O problema é classificado como sendo do tipo *Vehicle Routing Problem with Multiple Synchronization Constraints* (VRPMS) por Drex1, onde ao contrário de um *Vehicle Routing Problem* (VRP) clássico, existem múltiplos pontos de sincronização. A revisão bibliográfica efetuada revelou que não existe uma grande quantidade de informação publicada sobre o assunto. Contudo a revisão de literatura efetuada por Drex1 permitiu classificar os diferentes tipos de sincronização presentes no problema, nomeadamente a sincronização de movimentos, sincronização de operações exatas e a sincronização de operações com precedência.

A abordagem proposta nesta dissertação baseia-se no tipo de problemas *Home Health Care Routing and Scheduling Problem* (HHCRSP), onde é aplicada uma modelação orientada aos serviços que permite uma maior abstração do problema quando comparado com modelos de VRP convencionais. Na primeira fase do trabalho foi desenvolvido e implementado um modelo baseado numa abordagem orientada aos serviços, onde foram desenvolvidas inúmeras restrições e expressões lógicas de pré-processamento. As questões de pré-processamento estão diretamente ligadas com outros níveis de decisão e com a própria velocidade de computação do modelo. De seguida, o modelo foi testado para uma instância de pequena dimensão. Com os resultados destes testes é possível confirmar que o modelo otimiza as rotas consideradas e cumpre todos os aspetos de sincronização presentes no problema.

Numa fase posterior foram efetuadas análises complementares onde foram variados os números dos veículos. Desta análise conclui-se que, com o modelo desenvolvido, é possível realizar análises de natureza *trade-off* onde é possível escolher, no contexto do utilizador, o que é mais vantajoso: menos custos, intervalo temporal mais pequeno ou uma conjugação dos dois fatores.

O modelo desenvolvido foi incorporado num protótipo de um sistema de apoio à decisão em *Excel* que permite ao utilizador alterar os dados de *input*, executar o modelo e analisar resultados sem o ser necessário conhecimentos de programação.

No futuro, o modelo desenvolvido deve ser testado e ajustado para uma instância de dimensões próximas às reais. Devem ser desenvolvidas mais expressões e restrições de pré-processamento, com vista a acelerar a resolução do programa. O melhoramento do aspeto gráfico da interface desenvolvida também deve ser explorado pois permite uma melhor experiência de utilização.

Palavras-chave: planeamento de recolha e transporte, otimização, biomassa logística, programação matemática, planeamento de rotas, sincronização, HHCRSP, VRPMS

Abstract

The dissertation presented deals with a problem of planning and scheduling routes for collection and transport at the operational level in the biomass sector. The dissertation was developed in the INESC TEC institution within the scope of the research and development (R&D) EasyFlow project funded by the Fundação para a Ciência e Tecnologia (FCT).

Biomass has been ranked as one of the main sources of energy for achieving future energy security worldwide. The temporary and geographically fragmented availability of forest waste makes optimizing the entire collection and transportation process very important for reducing a company's costs.

The problem in question is classified by Drexl as being of type Vehicle Routing Problem with Multiple Synchronization Constraints (VRPMS), where unlike a classic VRP, there are multiple synchronization points. The bibliographic review on the subject revealed that there isn't a wide range of sources to discuss the problem in question. However, the Drexl literature review allowed the classification of the different types of synchronization present in the problem, namely synchronization of movements, synchronization of exact operations and synchronization of operations with precedence.

The proposal of resolution presented is based on the resolution of the problem Home Health Care Routing and Scheduling Problem (HHCRSP), where service-oriented modeling is applied, which allows a greater abstraction of the problem when compared to conventional VRP models.

In the first phase of the dissertation a model based on service-oriented modeling was developed and implemented, where a big number of logical constraints and expressions of preprocessing have been developed. Preprocessing issues are directly linked to other decision levels and to the model's own computing speed. Then the model was tested for a small instance. After analysing the results of these tests it is possible to confirm that the model optimizes the routes considered and fulfills all the synchronization aspects necessary for the problem.

At a later stage, complementary analyses were carried out where the numbers of the vehicles were varied and the results then analysed. From these analyses it is concluded that with the developed model it is possible to carry out analyses of a trade-off nature where it is possible to choose, in the context of the user, what is more advantageous: less costs, shorter time interval or a combination of the two factors.

The developed model was incorporated into a prototype of a decision support system in *Excel* which allows the user to change input data, execute the model, and analyse results without the need for programming knowledge.

In the future, the developed model must be tested and adjusted to an instance of dimensions close to a real one. Further preprocessing expressions and constraints should also be developed in order to speed up the computation of the program. The improvement of the entire graphic aspect of the developed interface should also be explored as it allows a better user experience.

Key-words: Optimization, planning, scheduling, logistics, synchronization, biomass, route planning and scheduling, VRPMS, HHCRSP

Agradecimentos

Em primeiro lugar, quero começar por agradecer à equipa do projeto *EasyFlow*, que me apoiaram em todas as dúvidas e confusões que tive no decorrer da dissertação: Vítor Mendes por ser o meu companheiro no INESC TEC e nas apresentações; Ricardo Soares por toda a ajuda na formulação e compreensão inicial do problema assim como pela ajuda contínua no desenvolvimento da dissertação; Maria João e Doutor Pedro Amorim pelos comentários construtivos nas apresentações; Doutor Jorge Pinho Sousa por todas as dicas e conhecimento partilhado durante a realização da dissertação e por fim à Doutora Alexandra Marques por ter tido um papel fundamental ao ter tirado um tempo todas as semanas para as reuniões que me mantiveram no ritmo de trabalho e pela disponibilidade incondicional apresentada.

Tenho ainda que agradecer a toda a minha família que me apoiou e me ajudou a chegar a este patamar. Pelas condições proporcionadas e pelo apoio em todos os momentos difíceis. Agradecer à minha prima Maria por me ter apresentado o mundo da faculdade e ao meu primo João por todos os momentos de descompressão.

Por fim agradecer a todos os meus amigos, sem eles nada disto seria possível. Agradecer ao pessoal dos cursos de Eletro, Mecânica, Química e Informática. Agradecer à minha amiga MJ que muitas tarde e noites me aturou e ajudou. Em especial agradecer ao francês Justin por ser o meu companheiro em todo o curso e a única pessoa que quer trabalhar de madrugada comigo. Não posso escrever o nome de toda a gente, mas ninguém que participou nesta minha aventura será esquecido.

Francisco Capa

*“The important thing is not to stop questioning.
Curiosity has its own reason for existing”*

Albert Einstein

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos e motivação	2
1.3	Descrição do problema	3
1.3.1	Problemas de decisão ao nível tático	4
1.3.2	Problemas de decisão ao nível operacional	5
1.4	Estrutura da dissertação	9
2	Revisão Bibliográfica	11
2.1	Introdução geral ao VRP	11
2.1.1	Métodos de resolução	12
2.2	Variantes de VRP	13
2.2.1	VRPTT	13
2.2.2	PDP	14
2.2.3	LTSP	15
2.2.4	SDVRP	15
2.2.5	Conclusão	16
2.3	Sincronização em problemas VRP	16
2.3.1	Diferentes tipos de sincronização	17
2.3.2	Sumário dos tipos de sincronização	26
2.4	<i>Home Health Care Routing and Scheduling Problem</i>	27
2.4.1	Formulação matemática	29
2.5	Enquadramento do problema no referencial de sincronização	30
3	Um novo modelo para sincronização em cadeias de abastecimento	33
3.1	Modelo matemático	33
3.1.1	Entidades do problema	33
3.1.2	Formulação matemática	36
3.1.3	Variáveis de decisão e função objetivo	38
3.1.4	Restrições	39
3.2	Extensão do modelo	39
3.2.1	Sincronização de operações em simultâneo	40
3.2.2	Sincronização de movimentos	41
3.2.3	Considerações adicionais	42
3.3	Caso de estudo	44
3.3.1	Geração da rede	46
3.3.2	Pré-processamento da rede	48

4	Resultados computacionais	51
4.1	Resolução do caso de estudo	51
4.2	Análises complementares	55
5	Interface gráfica de apoio	57
5.1	Desenvolvimento e apresentação	57
5.2	Manual de utilização	59
6	Conclusões e Trabalho Futuro	61
6.1	Conclusões e contribuições do trabalho desenvolvido	61
6.2	Trabalho Futuro	62
A	Código <i>Python</i> da implementação do modelo	65
A.1	Código do modelo	65
B	Código relativo à criação de instâncias	93
B.1	Código <i>Python</i>	93
C	Soluções alternativas para a instância apresentada	101
C.1	Solução com menos um veículo tipo A e uma estilhadora	101
C.2	Solução com menos um veículo do tipo B	102
C.3	Solução com menos um veículo de cada tipo	103
	Referências	105

Lista de Figuras

1.1	Possíveis fluxos de informação entre os dois níveis de decisão	3
1.2	Possíveis fluxos do transporte de mercadoria	6
1.3	Sincronização de movimentos	6
1.4	Operação de descarga da estilhadora	7
1.5	Operação de estilhagem	7
1.6	Fluxos de transporte de estilhadoras	8
1.7	Fluxos de transporte de resíduos	8
2.1	Sincronização de Tarefas	17
2.2	Sincronização de operações no domínio espacial	19
2.3	Sincronização de operações com precedência	19
2.4	Sincronização de operações exatas sem <i>offset</i>	21
2.5	Sincronização de operações exatas com <i>offset</i>	21
2.6	Sincronização de movimentos no depósito central	22
2.7	Sincronização de movimentos em rotas	23
3.1	Instância considerada	45
3.2	Exemplo de divisão de um ponto de recolha em nós falsos	47
3.3	Exemplo de divisão de um armazém em nós falsos	48
3.4	Exemplo de divisão de uma fábrica em nós falsos	48
4.1	Exemplo de sincronização de movimentos	53
4.2	Exemplo de sincronização de operações com precedência em veículos e em localizações	54
4.3	Exemplo de sincronização de operações simultâneas em localizações	54
5.1	Contribuição da interface gráfica.	57
5.2	Passos a ser executados pelo utilizador com a interface gráfica	58
5.3	Exemplo de alocação dinâmica das tabelas na interface gráfica.	58
5.4	Exemplo de distâncias calculadas automaticamente no ficheiro.	59

Lista de Tabelas

2.1	Artigos com sincronização de operações com precedência	20
2.2	Artigos com sincronização de movimentos em rotas	24
2.3	Sumário dos tipos de sincronização	27
2.4	Tipos de serviços e exemplos	28
2.5	Formulação matemática	29
2.6	Variáveis de decisão	30
2.7	Comparação de conjuntos	30
2.8	Tipos de serviços e exemplos no contexto de biomassa	31
3.1	Localizações existentes	34
3.2	Veículos existentes	34
3.3	Serviços, respetiva descrição e condições	35
3.4	Conjuntos aplicada ao contexto do problema	36
3.5	Parâmetros adicionais	37
3.6	Variáveis de decisão e descrição	37
3.7	Novo conjunto considerado	40
3.8	Novos parâmetros considerados	40
3.9	Dois novos serviços	41
3.10	Nova tabela de Serviços, respetiva descrição e condições	42
3.11	Novos serviços de exclusividade	43
3.12	Novos conjuntos considerados	43
3.13	Tabela final com todos os serviços	44
3.14	Quantidades disponíveis nos pontos de recolha	45
3.15	Capacidades máximas de cada armazém	45
3.16	Capacidades e procuras das fábricas	46
3.17	Diferentes custos para os diferentes veículos	46
3.18	Produtividade das estilhadoras nos diferentes pontos de recolha	46
3.19	Rotas previamente definidas	47
3.20	Conjuntos considerados no pré-processamento da rede	49
3.21	Novo parâmetro de rotas pré-determinadas	49
4.1	Solução final ordenada pelos tempos	52
4.2	Exemplo de sincronização de movimentos	53
4.3	Exemplo de sincronização de precedência nos veículos	53
4.4	Exemplo de sincronização de operações com precedência nas localizações	53
4.5	Exemplo de sincronização de operações simultâneas nas localizações	54
4.6	Três casos a analisar	55
4.7	Comparação com menos um camião tipo B	55

4.8	Comparação com menos um caminhão A e estilhadora	56
4.9	Comparação com menos um caminhão de cada tipo	56
C.1	Solução com menos um veículo tipo A e uma estilhadora	102
C.2	Solução com menos um veículo do tipo B	103
C.3	Solução com menos um veículo de cada tipo	104

Abreviaturas e Símbolos

ACO	Ant Colony Optimizaton
ALNS	Adaptative Large Neigghborhood Search
API	Application Programming Interface
ARP	Arc Routing Problem
CPU	Computer Processing Unit
FCT	Fundação para Ciências e Tecnologias
GA	Genetic Algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
HHCRSP	Home Health Care Routing and Scheduling Problem
INESC	Instituto de Engenharia de Sistemas e Computadores
INESC TEC	Instituto de Engenharia de Sistemas e Computadores - Tecnologia e Ciência
LTSP	Log-Truck Scheduling Problem
PDP	Pick-up and Delivery Problem
RAM	Random Access Memory
SDVRP	Split Delivery Vehicle Routing Problem
TS	Tabu Search
TSP	Travelling Salesman Problem
VRP	Vehicle Routing Problem
VRPMS	Vehicle Routing Problem with Multiple Synchronization Constraints
VRPTT	Vehicle Routing Problem with Trailers and Transhipments

Capítulo 1

Introdução

1.1 Enquadramento

A rápida utilização e esgotamento das reservas mundiais de combustíveis fósseis em conjunto com o seu prejudicial impacto ambiental associado à sua utilização tem levado a um aumento na utilização e produção de energia baseada em biocombustíveis. Tal como os combustíveis fósseis, os biocombustíveis também podem ser transformados numa grande variedade de produtos como papel, plástico, eletricidade, entre muitos outros.

Apesar da vantagem ambiental da utilização de biocombustíveis ainda existem alguns obstáculos que impedem que a biomassa se torne rapidamente o combustível mais utilizado. A sua descontínua disponibilidade temporal e geográfica em conjunto com a alta manutenção necessária e com os custos logísticos associados a uma cadeia deste género são os fatores que mais prejudicam a viabilidade económica para este tipo de combustíveis.

A biomassa tem sido classificada como uma das principais fontes de energia para a obtenção de uma futura segurança energética em combinação com uma grande redução das emissões de gases prejudiciais. Neste contexto, a biomassa é classificada como a fração biodegradável dos produtos e resíduos com origem biológica, incluindo fontes vegetais e animais. A grande expectativa à volta do papel da biomassa na produção de energia no futuro é justificada pois a indústria já está preparada para a utilização de biomassa, isto é, já existe um grande número de tecnologias preparadas para produzir energia através de biomassa. A redução das emissões prejudiciais pode até ser alcançada num curto prazo devido à disponibilidade da combinação de biomassa com carvão para a produção de energia nas atuais fábricas termoelétricas. Todos estes fatores em conjunto com o facto de a biomassa poder, ao contrário de outros tipos de energias renováveis, produzir vários tipos diferentes de energia (eletricidade, calor ou biocombustível) [1, 2] torna a biomassa uma solução muito atrativa como uma energia renovável.

A utilização de biomassa para a produção de energia tem vindo a aumentar significativamente nas últimas três décadas, sendo a Suécia um dos países que mais aposta na sua utilização como combustível. A título de exemplo, em 2013, a quantidade de energia produzida através de biomassa, na Suécia, atingiu praticamente 25% da totalidade de energia produzida [3].

Este tipo de combustível é utilizado numa grande variedade de setores, tendo, recentemente, sido bastante explorada a sua utilização em fábricas de aquecimento. Estas fábricas fornecem água quente, através de um sistema de tubos, a casas e apartamentos [4]. Com usos variados, mas especialmente focados no aquecimento, a indústria da biomassa é altamente sazonal, isto é, a procura por este tipo de energia aumenta significativamente nos meses mais frios do anos.

Apesar de muito promissor, o desenvolvimento de uma indústria de produção de energia forte e sustentável baseada em biomassa requer uma especial atenção no que diz respeito à cadeia logística. Normalmente uma cadeia logística deste género pode ser distinguida em três segmentos diferentes. Pode ser considerado um segmento superior que lida com as operações desde a produção até à conversão dos resíduos em biomassa, um segmento médio que considera apenas o processo de conversão e por fim o segmento inferior que lida com as operações de transporte para os consumidores ou com o armazenamento dos produtos finais [5, 6]. Esta dissertação foca-se muito neste aspeto de desenvolvimento de uma cadeia logística otimizada.

O problema da gestão integrada das operações de recolha e transporte de biomassa, assim como a sua sincronização, reside numa complexidade de obstáculos que tem em conta vários aspetos. De uma forma geral, este é um problema típico de gestão de cadeias de abastecimento, uma vez que existem várias fontes de matéria prima distintas com disponibilidade conhecida, vários consumidores com procura conhecida, que se distribuem ao longo de uma rede logística que inclui vários pontos intermédios destinados ao armazenamento também podem existir diferentes tipos de matéria prima onde poderão ser considerados diferentes períodos temporais [7].

1.2 Objetivos e motivação

Esta dissertação tem como objetivo a aplicação de técnicas de investigação operacional e de sistemas de apoio à decisão ao planeamento e controlo das operações de destroçamento de resíduos florestais e o seu respetivo carregamento sincronizado para camiões de transporte para centrais de biomassa.

A disponibilidade temporária e geograficamente fragmentada de resíduos florestais leva a um aumento de importância nas decisões sobre a produção, armazenamento e transporte dos mesmos. É então necessário encontrar uma solução que minimize os custos enquanto produz os resultados esperados. Este problema está a ser investigado no INESC TEC no âmbito do projeto *EasyFlow*. O INESC TEC é uma instituição privada sem fins lucrativos associado e coordenado pelo INESC Porto, que tem como parceiros a Universidade do Porto, o INESC e o Instituto Politécnico do Porto. O projeto *EasyFlow* tem como objetivo aumentar a sustentabilidade de cadeias de abastecimento de base florestal através de ferramentas computacionais e tecnologias sensoriais para melhorar a eficiência dos recursos e permitir o planeamento logístico sustentável e colaborativo.

O principal objetivo do trabalho é o desenvolvimento de um modelo matemático que permita a sincronização das operações mencionadas assim como a sua implementação e execução com recurso a uma linguagem de modelação e a um *solver* comercial (*Gurobi*). A abordagem proposta

nesta dissertação tem como propósito a redução de custos de transporte e a otimização do processo de recolha, assim como a sua integração com o processo de transporte.

1.3 Descrição do problema

Num problema deste género podem ser considerados pelo menos dois níveis de decisão, tático e operacional. O nível tático está relacionado com decisões de médio a longo prazo, enquanto o nível operacional se relaciona com decisões em períodos temporais diários ou semanais. A figura 1.1 demonstra uma possível representação de dois níveis de decisão diferentes mas interligados entre si.

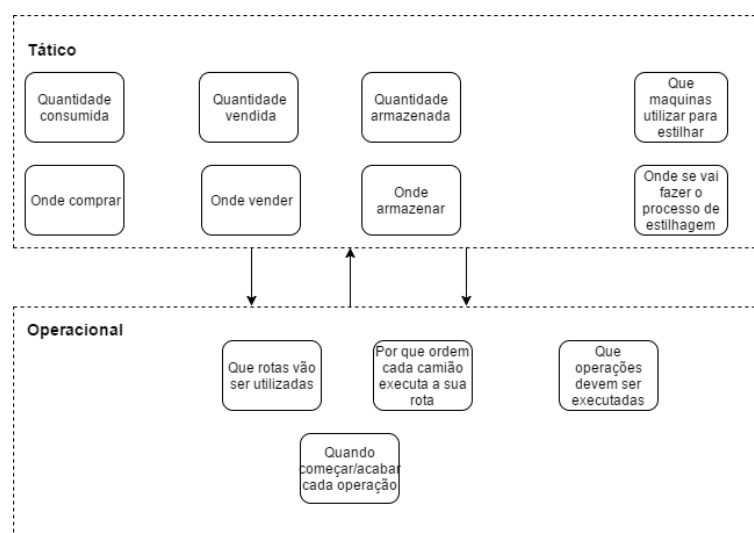


Figura 1.1: Possíveis fluxos de informação entre os dois níveis de decisão

Na figura 1.1 estão representados apenas alguns exemplos de decisões que podem ser tomadas nos respetivos níveis. Como é possível observar existe uma ligação entre as decisões que são tomadas. Por exemplo, a nível tático, pode ser tomada a decisão de não utilizar um certo fornecedor por falta de qualidade nos resíduos florestais, fazendo assim com que a nível operacional não seja possível utilizar os nós e as rotas que representam esses fornecedores. A mesma situação pode ocorrer inversamente, se uma rota entre um fornecedor e um consumidor ou armazém seja demasiado cara e aumente significativamente os custos, a nível operacional pode ser tomada a decisão de não utilizar o fornecedor ou consumidor.

No problema apresentado foi tomada uma decisão tática de apenas ser possível efetuar o processo de estilhagem nos pontos da cadeia onde os resíduos são recolhidos. As estilhadoras são considerados veículos não autónomos que necessitam de outro veículo para se movimentarem entre os diferentes pontos de recolha. São então considerados três tipos distintos de veículos: as estilhadoras, os veículos destinados ao transporte das estilhadoras e os veículos de transporte de resíduos florestais.

1.3.1 Problemas de decisão ao nível tático

A nível tático são tomadas decisões de alto nível a médio e longo prazo que afetam as decisões que vão ser tomadas mais tarde a um nível operacional. Neste nível podem ser distinguidas decisões no âmbito da recolha de resíduos, do armazenamento dos mesmos ou até na escolha dos consumidores.

1.3.1.1 Armazéns

Antes de o transporte para o consumidor ser efetuado, a matéria prima pode ser conservada em armazéns. Estes terminais são utilizados para equilibrar a variação na procura e na disponibilidade destes materiais.

Os armazéns também são utilizados como uma forma de aumentar o valor energético da biomassa. Ao armazenar os resíduos florestais obtidos, a humidade presente nos mesmos vai diminuir, aumentando assim o seu valor energético e consequentemente o seu valor monetário.

No mesmo armazém são considerados diferentes custos de armazenamento dependendo das condições de armazenamento; enquanto o armazenamento de resíduos já estilhados é mais caro, uma vez que necessita de uma superfície dura e proteção contra a chuva, o armazenamento de resíduos não estilhados pode ser feito em qualquer superfície. Assim, alguns armazéns possuem a capacidade de estilhar o material no local enquanto outros apenas possuem essa capacidade quando as máquinas necessárias para a operação se deslocam ao local. O local onde é feita a estilhaagem pode ser o armazém ou o próprio ponto de recolha dependendo dos custos logísticos associados. Em alguns casos, este tipo de terminais estão localizados perto de portos marítimos e são apenas utilizados quando os resíduos florestais são importados e não recolhidos.

1.3.1.2 Fábricas

Neste caso as fábricas são consideradas os consumidores da matéria prima que vai ser recolhida. Devem, por isso, ser consideradas as flutuações na procura que estas fábricas sofrem devido às condições climáticas em cada região. A título de exemplo pode ser considerado o caso da Suécia, em que, durante o Inverno, a procura é significativamente elevada para este tipo de produção de energia, o que contrasta com a procura praticamente inexistente que se verifica no Verão.

Esta procura é normalmente especificada em MWh para cada período temporal considerado, e é então necessário perceber como converter os volumes de resíduos estilhados em energia produzida. Diferentes tipos de resíduos produzem diferentes quantidades de energia. Para efetuar esta conversão são utilizados vários fatores, como por exemplo, a humidade existente nos resíduos.

1.3.1.3 Resíduos florestais

A biomassa é normalmente obtida através de resíduos florestais ou como um subproduto de outro tipo de atividades relacionadas com o corte de madeira. Estes subprodutos podem ser transportados diretamente para as fábricas, uma vez que não necessitam de mais nenhum processo

adicional.

Por resíduos florestais entende-se tudo que é descartado noutro tipo de processos semelhantes, como folhas, galhos, cascas e outras partes de madeira que não são aproveitadas. Estes resíduos florestais requerem um processo adicional onde a madeira é estilhada para mais tarde ser convertida em energia. O processo normalmente utilizado consiste em recolher estes resíduos, transportá-los para um armazém e aguardar algum tempo para que a matéria prima perca humidade e assim ganhe valor. Desta forma pode ser mais tarde estilhada e transportada para os consumidores.

1.3.2 Problemas de decisão ao nível operacional

O problema discutido nesta dissertação consiste na sincronização entre as operações de recolha e transporte de biomassa, aplicado mais tarde a uma empresa na Finlândia que opera nesta área. Trata-se então de um problema ao nível operacional que lida com decisões de planeamento das operações de estilhagem e transporte que precisam de ser sincronizadas, isto é, têm de acontecer no mesmo local e no mesmo período temporal. O processo de estilhagem é efetuado por máquinas especializadas que são transportadas para um ponto de recolha e mais tarde movidas para outro ponto. Este processo é efetuado diretamente para os camiões de transporte. As cargas são então transportadas para armazéns ou consumidores previamente definidos.

Nas próximas secções são apresentadas as decisões a nível operacional associadas ao problema em estudo e a alguns pontos da cadeia logística.

1.3.2.1 Transporte

Considerando, numa primeira fase, apenas a componente de transporte de biomassa, é necessário ter em conta a existência de múltiplos fornecedores (zonas de recolha) e múltiplos consumidores (fábricas) assim como a existência de armazéns com capacidade limitada, que são considerados um ponto intermédio na cadeia logística.

O transporte é normalmente feito por diferentes empresas com recurso a vários tipos de camiões. Por exemplo, existem diferentes tipos de camiões para transportar matéria prima já estilhada e matéria prima por estilhar. Assim, a principal decisão a ser tomada consiste na escolha do destino dos resíduos já estilhados, isto é, se devem ser armazenados ou transportados diretamente para o consumidor. Por sua vez, os resíduos que ainda não foram estilhados no momento da sua recolha têm de ser transportados para terminais ou fábricas com capacidade para efetuar este processo.

Os custos de transporte correspondem à maior parte dos custos da cadeia logística, aproximadamente um terço dos custos totais. Na figura 1.2 estão representados por setas alguns dos fluxos de mercadoria existentes no contexto de transporte de biomassa. É necessário ter em conta que a figura 1.2 é um caso muito simples deste tipo de problema, contendo apenas um ponto de recolha, dois armazéns e duas fábricas. De notar que neste tipo de problema o transporte é uma decisão a nível operacional pelo facto de existirem múltiplos fluxos possíveis para as diferentes localizações.

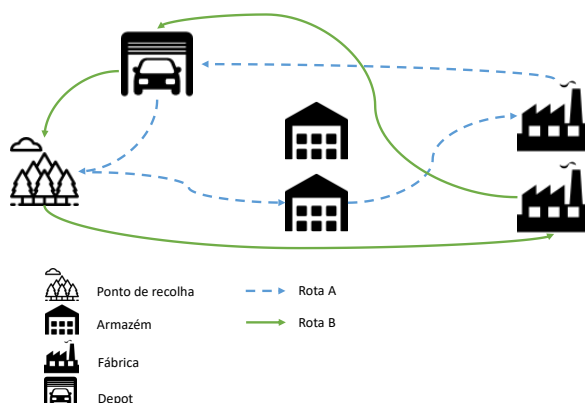


Figura 1.2: Possíveis fluxos do transporte de mercadoria

1.3.2.2 Sincronização de movimentos

Para além das decisões associadas ao transporte num *Vehicle Routing Problem* (VRP) é necessário considerar uma sincronização de movimentos, isto é, como as estilhadoras são veículos móveis não autónomos, necessitam de outros veículos autónomos para se movimentarem. É então necessário sincronizar o momento em que as estilhadoras começam e acabam as suas rotas com os momentos em que os veículos destinados ao transporte de estilhadoras começam e acabam a sua rota.

Na figura 1.3 está representada a sincronização de movimentos desejada neste problema, com um pequeno desfasamento nas rotas para serem distinguíveis.

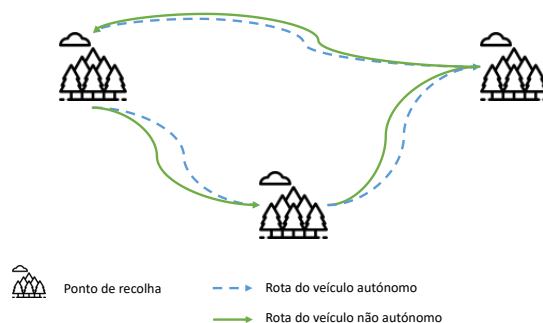


Figura 1.3: Sincronização de movimentos

1.3.2.3 Sincronização de operações

Para otimizar o processo de estilhagem e os processos de carga e descarga de resíduos é necessário considerar todas as operações e os momentos em que ocorrem. Com o objetivo de minimizar os tempos de inatividade em cada ponto da cadeia é necessária uma sincronização de operações.

Nas figuras 1.4 e 1.5 estão representadas as duas operações necessárias para o processo de recolha e estilhagem de biomassa. É possível observar que, numa primeira fase, é necessário descarregar a estilhadora dos veículos destinados ao transporte da mesma, sendo depois possível proceder para a estilhagem dos resíduos. O processo de estilhagem é feito diretamente para o veículo que vai transportar o material já estilhado para as fábricas ou armazéns. É então necessária uma sincronização de operações com precedência.

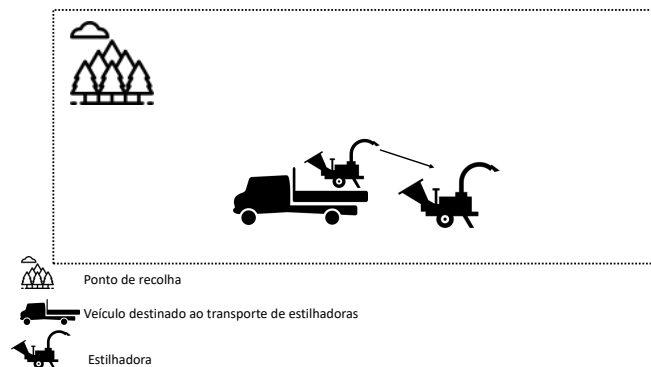


Figura 1.4: Operação de descarga da estilhadora

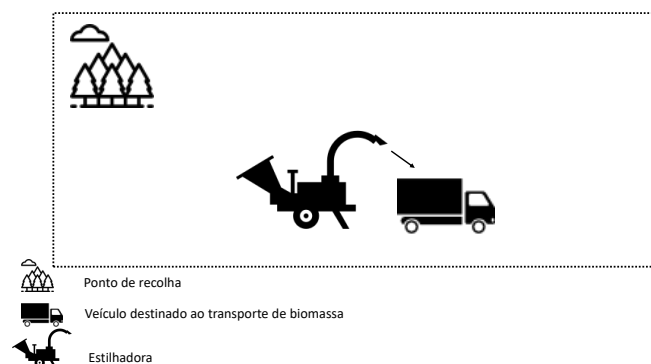


Figura 1.5: Operação de estilhagem

1.3.2.4 Sincronização entre recolha e transporte

Para otimizar o processo de estilhagem dos resíduos florestais foi considerado que este processo é efetuado diretamente para os camiões destinados ao transporte dos resíduos. Assim, é então necessária uma sincronização entre as rotas de recolha e de transporte. Por sincronização de operações de recolha e transporte entende-se a sincronização dos momentos em que os veículos destinados ao transporte de matéria prima chegam aos pontos de recolha e o momento em que a estilhagem da matéria prima é feita. O objetivo desta sincronização passa pela minimização dos tempos em que os veículos estão inativos à espera dos produtos estilhados ou vice-versa. Numa

situação ideal os tempos de inatividade e de espera dos veículos seriam nulos, levando ao máximo aproveitamento dos recursos existentes. No entanto é importante realçar que tal situação pode conduzir à utilização excessiva de um número de veículos, que se poderá revelar economicamente inviável.

No problema em questão é considerado que a estilhagem é feita nos pontos de recolha e as máquinas destinadas a este processo podem ser movidas entre os diferentes pontos de recolha, assim o problema pode ser visto como um problema de sincronização de dois *Vehicle Routing Problem* distintos. Nas figuras 1.6 e 1.7 estão representados os dois VRP's a considerar. O problema está na sincronização da rota A com a rota B, isto é, a rota A (recolha e estilhagem) representa a movimentação das estilhadoras entre os pontos de recolha e a rota B representa o transporte da biomassa entre os pontos de recolha e os diferentes armazéns ou fábricas.

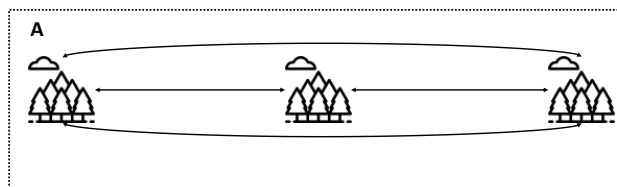


Figura 1.6: Fluxos de transporte de estilhadoras

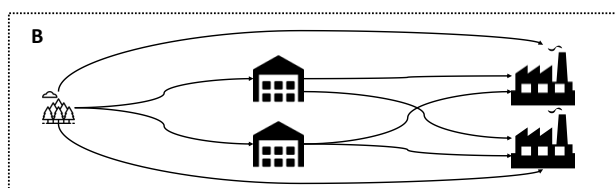


Figura 1.7: Fluxos de transporte de resíduos

De notar que nas rotas de recolha (rota A) está implícita uma sincronização das operações mencionadas anteriormente assim como a sincronização de movimentos associada ao transporte das estilhadoras.

1.4 Estrutura da dissertação

Para além da introdução, esta dissertação contém mais 5 capítulos. No capítulo 2, é descrito o estado da arte e são apresentados trabalhos relacionados. No capítulo 3, é descrito todo o modelo implementado, assim como as questões de sincronização e de pré-processamento. No capítulo 4 são apresentados e discutidos os resultados da otimização do problema, assim como são realizadas análises complementares. No capítulo 5 é descrita uma interface gráfica que foi desenvolvida em *Excel*. No capítulo 6 são discutidos todos os resultados obtidos e tiradas conclusões acerca do modelo e da abordagem proposta.

Capítulo 2

Revisão Bibliográfica

Neste capítulo serão apresentados os resultados da revisão bibliográfica efetuada sobre o tema *Vehicle Routing Problem* e a consequente sincronização de vários VRP. Como o foco principal desta dissertação é a sincronização, a revisão está orientada para a sincronização em VRP. Contudo numa fase inicial foi feita uma pequena revisão bibliográfica a problemas VRP, para esta ser tomada como base para a revisão da sincronização em VRP.

2.1 Introdução geral ao VRP

O VRP é uma generalização do problema de otimização combinatória *Travelling Salesman Problem* inicialmente formulado por Dantzig e Ramser [8]. Na sua versão mais simples, o VRP tem como objetivo determinar as rotas com o custo ótimo, para um ou mais veículos, para entrega ou recolha de mercadoria ao longo de uma rede de distribuição.

Neste tipo de problema as requisições de transporte de produtos estão normalmente concentradas em pontos específicos (nós) de uma rede de estradas. No caso de as requisições de transporte estarem dispersas ao longo de arcos, isto é, ruas ou segmentos de uma rua, o problema é então do tipo Arc Routing Problem (ARP). Por exemplo, serviços relacionados com a limpeza de ruas no Inverno ou até com a entrega de correio podem ser considerados do tipo ARP.

Numa formulação deste género os elementos da rede de distribuição apresentados têm a si associados papéis de alta importância para a resolução de um problema de planeamento de rotas, nomeadamente:

- Vértices ou **nós de origem**, podendo ser considerados os **fornecedores** de uma cadeia de abastecimento. Estes nós são normalmente considerados armazéns onde está localizada a oferta de um determinado produto.
- Vértices ou **nós de destino**, podendo ser considerados os **consumidores** de uma cadeia de abastecimento. Neste tipo de nós está então localizada a procura de um determinado produto.

- **Arcos** entre vértices, em que são consideradas as **rotas** existentes entre os diferentes pontos de interesse da cadeia de abastecimento. Têm a si associado um custo, que depende da distância entre nós e de outros dados como o preço do combustível, e ainda o tempo que é necessário para percorrer um arco.
- Em alguns casos é também considerado um **depósito central** que atua como início e fim de todas as rotas.
- **Veículos**, sendo os elementos que efetuam o transporte de produtos entre os vários pontos da rede de distribuição e podem também estar sujeitos a restrições de capacidade ou de tempo, isto é, em casos reais existe um limite de capacidade para todos os veículos e a cada veículo está sempre associado um condutor que tem um número de horas de trabalho fixas que têm de ser tidas em conta.

Para a formulação de um VRP têm de ser consideradas **variáveis de decisão**, cujo formato pode ser alterado consoante os requisitos do problema [9]. São consideradas **restrições**, que alteram a natureza do problema, por exemplo, em [10, 11] são adicionadas restrições de capacidade. Finalmente, tem de ser considerada uma **função objetivo** que define, em qualquer problema de otimização combinatoria, que valor é que se pretende maximizar ou minimizar.

Este tipo de planeamento de rotas e a sua otimização é um problema do tipo NP-difícil e requer uma certa atenção na escolha do método de resolução, pois para casos reais com grandes quantidades de dados, o tempo de resolução computacional pode atingir valores demasiado elevados [12]. É importante referir a relevância económica da resolução deste tipo de problemas com recurso a *softwares* de modelação e otimização. Ao utilizar ferramentas computacionais para determinar as soluções ótimas é possível alcançar uma significativa redução nos custos e uma melhor utilização das frotas de camiões consideradas [13].

2.1.1 Métodos de resolução

A maior parte dos problemas do tipo VRP e as suas variantes são considerados do tipo NP-difícil, significando isto que para se encontrar uma solução em tempos de computação razoáveis é necessária a utilização de heurísticas e meta-heurísticas, estes métodos de resolução são considerados **métodos aproximados**. Contudo, em instâncias de pequena dimensão, são utilizados **métodos exatos** para calcular soluções ótimas do problema.

Como **métodos exatos** entendem-se todos os métodos que apenas param quando encontram a melhor, ou uma das melhores, soluções para o problema. Para ser possível utilizar este tipo de métodos não podem ser consideradas grandes quantidades de dados, isto é, apenas problemas com pouca dimensão podem ser resolvidos em tempos razoáveis. Em [14] pode ser encontrada uma revisão bibliográfica sobre métodos de resolução.

Alguns exemplos de métodos de resolução exatos utilizam *Column Generation* [15, 16, 17] e *Branch and Bound* de onde surgiram as variantes *Branch and Cut* e *Branch and Price* [18].

Devido à natureza Np-difícil da maior parte dos problemas do tipo VRP e suas variantes é necessário considerar diferentes **métodos de resolução** que não utilizem demasiados recursos computacionais e não demorem demasiado tempo a encontrar a solução ótima.

Os métodos de resolução aproximados podem ser classificados em dois grandes grupos, as heurísticas e as meta-heurísticas. Por métodos heurísticos entendem-se métodos que são dependentes do problema e requerem muitos ajustes para cada caso, por outro lado uma meta-heurística é um método de resolução aproximado independente do problema em questão, que requer apenas alguns ajustes mínimos dos parâmetros intrínsecos ao problema, e pode ser aplicado numa grande variedade de casos.

Uma heurística pode levar a uma solução ótima local e ignorar uma melhor solução que possa existir, enquanto uma meta-heurística leva na maioria dos casos a um ótimo global do problema. Por sua vez, uma das formas de classificar meta-heurísticas é de acordo com o número de soluções utilizadas em simultâneo.

Assim, as meta-heurísticas baseadas em populações mantêm e melhoram várias soluções ao mesmo tempo. Como exemplo podem ser considerados *Ant Colony Optimization* (ACO) [19, 20] e *Genetic Algorithm* (GA). As meta-heurísticas individuais focam-se exclusivamente na otimização de uma solução, ou seja, partem de uma solução inicial em vez de uma população de soluções iniciais. Como exemplo podem ser considerados os algoritmos *Tabu Search* (TS) [21, 22, 23] e *Greedy Randomized Adaptive Search Procedure* (GRASP) [24, 25]. Contudo, o algoritmo *Adaptive Large Neighborhood Search* (ALNS) é dos métodos heurísticos aquele que mostra mais eficiência e eficácia na procura das soluções [26].

2.2 Variantes de VRP

Na prática os problemas do tipo VRP podem ser modificados tornando-os mais complexos e com maiores aplicações através da aplicação de novas generalizações à formulação do problema. Após uma breve introdução às variantes mais utilizadas de VRP na secção de sincronização, vão então ser apresentadas as variantes que mais se relacionam com o caso em estudo.

2.2.1 VRPTT

A variante *Vehicle Routing Problem with Trailers and Transshipments* pode-se inserir nas classes de sincronização de operações e de sincronização de movimentos. Esta variante foi introduzida por Drexl [27] no contexto da recolha de leite em várias quintas.

O VRPTT enquadra-se no foco desta dissertação por ser classificada por Drexl como uma representação genérica da classe de *Vehicle Routing Problems with Multiple Synchronization Constraints* (VRPMS). Ao contrário de um VRP clássico, onde a sincronização só é necessária no que diz respeito a que veículo vai visitar cada cliente, num VRPMS já é necessário garantir a sincronização entre cargas, períodos temporais e localizações.

Num VRPTT existe um conjunto de clientes com uma determinada procura e um conjunto de veículos heterogéneos. Para recolher a matéria prima são usados dois tipos de veículos, os veículos

autónomos e os não autónomos. Um veículo autónomo pode-se movimentar no espaço e no tempo livremente enquanto um veículo não autónomo só o pode fazer se for puxado por outro veículo. Para além de num problema deste tipo também existirem depósitos centrais, podem também ser considerados locais destinados a efetuar as trocas entre os veículos autónomos e não autónomos [28].

O problema consiste em determinar rotas para os veículos autónomos e as rotas para os veículos não autónomos de forma a minimizar os custos de transporte. O aspeto mais importante nesta variante é a sincronização necessária das rotas dos dois tipos de veículos, isto é, as rotas têm de estar sincronizadas no que diz respeito ao tempo, local e carga.

2.2.2 PDP

Os *Pickup and Delivery Problems* (PDP) constituem uma classe importante de *Vehicle Routing Problems* em que objetos ou pessoas têm de ser transportados entre um ponto de origem e o seu destino. Esta variante insere-se na sincronização de operações, carga e tarefas. Este tipo de problemas têm sido estudados há mais de trinta anos, com origem em contextos como a logística, serviços ambulatoriais e robótica.

Em PDP's, uma frota de veículos com origem num depósito central é utilizada para completar um série de pedidos. Um pedido consiste no transporte de mercadoria a partir de uma localização específica, onde a mercadoria é carregada, até outra localização, onde a mercadoria é descarregada [29]. É atribuído um período temporal a cada localização de recolha ou de entrega, especificando assim o intervalo de tempo que o serviço deve começar. Um veículo pode ou recolher ou entregar a quantidade total disponível ou necessária em cada localização.

Em suma, PDP's consistem na construção de um número limitado de rotas com o objetivo de minimizar o custo total das mesmas, sujeitas às seguintes restrições:

- Todos os pedidos de recolha e entrega têm de ser cumpridos.
- Não podem existir trocas de matéria prima nas localizações não adequadas.
- A carga de um veículo nunca pode exceder a sua capacidade máxima.

Em [30] é sugerida uma classificação de PDP's de acordo a três campos: estrutura, visitas e veículos.

No primeiro campo, denominado estrutura, são especificadas as quantidades de nós de origem e nós de destino que a matéria prima pode ter. A título de exemplo, em problemas muitos-para-muitos (M-M), qualquer nó pode servir como destino ou origem para a mercadoria, enquanto em problemas um-para-muitos-para-um (1-M-1), a mercadoria está, numa fase inicial, disponível no depósito central tendo como destino os consumidores, e a mercadoria que está disponível nos consumidores tem como destino o depósito central.

O segundo campo, fornece informação sobre como as operações de recolha e entrega são efetuadas nos nós representados pelos consumidores. A título de exemplo, é utilizada a notação PD

para indicar que cada consumidor só pode ser visitado uma vez para uma operação de recolha ou entrega, enquanto a notação P-D indica que as duas operações podem ser executados em conjunto ou em separado. Por fim, o último campo diz respeito ao número de veículos a utilizar na solução.

2.2.3 LTSP

O *Log-Truck Scheduling Problem* é a variante de VRP que serve como base para o problema apresentado nesta dissertação. Nesta variante a matéria prima a considerar consiste em diferentes tipos de troncos de madeira cortada, sendo que cada tipo de madeira tem um valor e uma qualidade diferente. O objetivo principal é a minimização dos custos de transporte dos troncos de madeira desde as florestas até aos consumidores.

O LTSP consiste na construção de um plano de transporte num horizonte temporal definido, em que são entregues o número de cargas necessárias em todos os pontos da cadeia de abastecimento sem violar as restrições dos fornecedores e dos consumidores com um custo de transporte mínimo. Associada às rotas, é necessário também tomar uma decisão sobre a quantidade de madeira que se vai transportar entre fornecedor e consumidor, de maneira a minimizar ou maximizar os lucros da cadeia de abastecimento [31].

A função objetivo de um problema LTSP normalmente inclui os custos de atividades não produtivas e os custos de transporte de uma carga desde um fornecedor até um consumidor em cada camião. Por atividades não produtivas entendem-se todas as atividades que não adicionam valor ao produto no ponto de vista do consumidor, por exemplo, neste caso uma atividade não produtiva é o tempo de espera que os camiões podem ficar quando estão à espera de carregar ou descarregar os seus produtos, admitindo que apenas um camião pode executar as operações de carga e descarga em cada localização, fazendo assim os outros esperar. Para uma boa modelação deste tipo de problema este parâmetro não pode ficar esquecido, tendo em conta que a maior parte das vezes que existe um tempo de espera este acarreta custos elevados para uma empresa de transporte [32].

O LTSP está relacionado com os problemas do tipo VRP, mais particularmente com o *Pick-up and Delivery Problem with Time Windows*, com a particularidade de apenas serem consideradas cargas completas, isto é, apenas são consideradas cargas com a mesma quantidade que a capacidade máxima dos camiões de transporte.

2.2.4 SDVRP

O *Split Delivery Vehicle Routing Problem* (SDVRP) distingue-se das classes normais de VRP por permitir que cada consumidor de matéria prima seja servido por um ou mais veículos. Esta classe insere-se na sincronização de cargas na medida em que é necessário tomar decisões no que diz respeito à quantidade de carga em cada veículo. Nos casos em que a procura de um certo consumidor é superior à capacidade de transporte dos veículos é óbvio que a encomenda tem que ser dividida entre dois ou mais veículos, porém, mesmo nos casos em que a procura não ultrapassa

a capacidade dos veículos, a divisão das encomendas pode levar a grandes reduções de custos [33].

Um exemplo de aplicação a casos reais pode ser encontrada em [34].

No que diz respeito aos métodos de resolução, a literatura foca-se principalmente em métodos heurísticos para resolver SDVRP e as suas variantes. A primeira heurística desenvolvida e aplicada em SDVRP foi um método de procura local, método este que envolve dois tipos de operadores de procura local [33]. Em [35] foi desenvolvida uma heurística que combina um modelo baseado em Programação Inteira Mista e um algoritmo de *record-to-record*. Em [36] é proposto um método heurístico de *Tabu Search* para resolver um problema do tipo k-SDVRP em que a capacidade de cada veículo, a procura dos consumidores e as quantidades transportadas para os consumidores são valores inteiros.

Mais recentemente foi proposta uma resolução alternativa em [37] para o SDVRP, onde a capacidade de cada veículo e as procuras de cada consumidor não são necessariamente números inteiros, o número de veículos não está limitado para o menor número possível e as procuras dos consumidores podem exceder as capacidades dos veículos.

2.2.5 Conclusão

Apesar da existência de uma grande variedade de variantes VRP, nenhuma se relaciona completamente com o problema de gestão integrada das operações de transporte e recolha de biomassa. O maior desafio passa por incluir vários tipos de sincronização na mesma variante de VRP, o que não acontece nas apresentadas. Normalmente apenas é considerado um tipo de sincronização em cada variante de VRP. No problema em questão são necessário pelo menos dois tipos distintos de sincronização, a sincronização de movimentos em rotas e a sincronização de operações com precedência.

De um modo geral, as formulações de VRP que incluem sincronização de movimentos, focam-se nessa mesma sincronização e ignoram outros aspetos do problema, sendo impossível conciliar a sincronização de operações com movimentos.

Foi então necessária uma abordagem diferente para uma boa modelação e generalização do problema. A primeira hipótese passa pela sincronização de dois VRP's, algo que ainda não foi desenvolvido com detalhe na literatura. A segunda hipótese é baseada no *Home Health Care Routing and Scheduling Problem* (HHCRSP) onde apenas são considerados serviços. Esta última foi a abordagem escolhida para a resolução do problema desta dissertação. Nas secções seguintes será apresentada uma pequena explicação sobre o HHCRSP assim como o seu enquadramento no problema.

2.3 Sincronização em problemas VRP

O foco desta dissertação reside numa classe específica dos problemas do tipo VRP, *Vehicle Routing Problem with Multiple Synchronization Constraints* (VRPMS), que tem sido alvo de bastante interesse nos últimos anos.

Trata-se de uma variante bastante abrangente que lida com múltiplos tipos de sincronização, desde sincronização de tarefas e operações numa determinada rota, à sincronização de movimentos e carregamentos. A maioria da literatura disponível sobre este assunto apenas foi publicada nos últimos anos, sendo uma grande parte da autoria de Michael Drexl [38].

2.3.1 Diferentes tipos de sincronização

Em VRP sem sincronização os veículos considerados são independentes uns dos outros, isto é, uma mudança numa rota não afeta qualquer outra rota que esteja a ser executada por outro veículo. Em problemas do tipo VRPMS uma mudança numa rota pode afetar as outras rotas, sendo que no caso mais extremo, uma mudança numa rota pode originar a impossibilidade de execução de todas as outras rotas. Este é um problema de interdependência e requer diferentes tipos de sincronização.

2.3.1.1 Sincronização de tarefas

O objeto principal em problemas VRP são as tarefas. Uma tarefa é algo que tem que ser cumprido e requer zero ou mais unidades de capacidade para ser executada. Uma tarefa pode consistir na recolha de matéria prima num determinado local, no transporte de matéria prima, na execução de um serviço, entre muitos outros.

Um veículo é um objeto móvel, autónomo ou não, que fornece zero ou mais unidades de capacidade e pode ou não ser utilizado na execução de uma tarefa.

Por sincronização de tarefas entende-se a decisão de qual o veículo que vai executar cada tarefa. Esta sincronização é o aspeto que diferencia os problemas do tipo TSP (*Travelling Sales Problem*, onde é considerado um único veículo) de problemas do tipo VRP (onde são considerados múltiplos veículos), pelo que no TSP esta sincronização não é considerada.

Na figura 2.1 está representada uma situação que contém este tipo de sincronização. Neste caso os veículos 1 e 2 são idênticos e ambos podem executar a tarefa 1, resta então a decisão de qual veículo é que vai executar a tarefa.

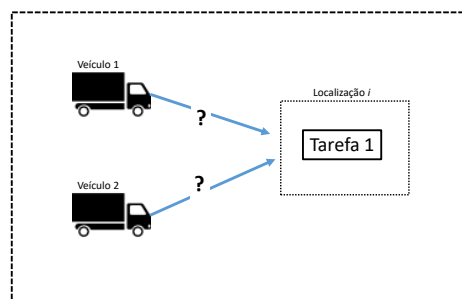


Figura 2.1: Sincronização de Tarefas

2.3.1.2 Sincronização de operações

Uma operação é algo que pode ou tem de ser executada por um veículo quando chega a um determinado local da sua rota, para que assim seja possível a execução de outras tarefas.

Sincronização de operações diz respeito à sincronização das operações dos diferentes veículos em iguais ou diferentes localizações, tendo em atenção ao instante em que os veículos executam as suas operações nos locais destinados. Assim, a sincronização de operações decide os domínios espaciais e temporais em que as tarefas são executadas.

Este tipo de sincronização pode induzir janelas temporais dinâmicas. Estas janelas são períodos temporais que dependem fortemente de outros e, por conseguinte, estão constantemente em mudança. Assim, a computação de planeamentos e rotas para um veículo sem ter em conta todos os outros não é considerada sincronização de operações. O *Vehicle Routing Problem with Trailers and Transhipments* [28] é um problema particular deste tipo de sincronização onde as trocas de mercadoria só são possíveis se ambos os veículos passivo e ativo estiverem presentes no local onde vai ser efetuada a troca. A janela temporal dinâmica em que o veículo passivo, envolvido na troca de matéria prima, pode começar a executar a sua tarefa, isto é, receber matéria prima, depende do momento em que o veículo ativo chega ao local e vice-versa. Este momento em que o veículo ativo ou passivo chega ao local não é conhecido mas sim calculado no decorrer do algoritmo.

O problema de sincronização de operações pode então ser definido na medida em que o *offset* temporal, isto é, o tempo que passa entre a execução de uma determinada operação por um determinado veículo num determinado local e a execução de outra tarefa executada por diferentes intervenientes num local diferente, tem que estar entre um intervalo de valores finito entre zero e um limite estabelecido.

No que diz respeito ao aspeto temporal das operações podem ser considerados três tipos diferentes de sincronização, onde o *offset* δ tem de estar dentro do intervalo $[a, b]$, $a \leq b$.

- **Sincronização de operações no domínio espacial** - Este é o caso em que num modelo ou algoritmo, o aspeto temporal é completamente ignorado e apenas são consideradas as sincronizações espaciais necessárias.

Por exemplo, num VRPTT as trocas de mercadoria só podem ser executadas quando os veículos ativo e passivo estão no mesmo local, se no algoritmo não for considerado o facto de os veículos terem de visitar os pontos de troca de mercadoria ao mesmo tempo o problema é apenas resolvido com sincronização espacial. Neste caso os veículos ativos e passivos podem passar pelo mesmo local mas não ao mesmo tempo e este aspeto de sincronização temporal é realizado manualmente por um operador especializado.

Na imagem 2.2 está representada este tipo sincronização na medida em que para ser executada, a operação 1, requer que ambos os veículos visitem a mesma localização mas não é especificado o horizonte temporal para nenhum dos veículos.

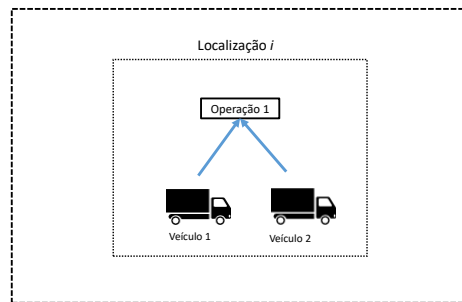


Figura 2.2: Sincronização de operações no domínio espacial

Com este tipo de sincronização surge uma classe de VRP importante, a classe *N-echelon* VRP. Esta variante foi introduzida pela primeira vez em 2008 [39]. Uma revisão de literatura sobre *N-echelon* VRP revela que as heurísticas utilizadas na resolução deste tipo de problemas se baseiam na decomposição por estados como ideia central. Quando não se valorizam os períodos temporais, os estados podem ser considerados sequencialmente ou iterativamente, podendo assim obter soluções de alta qualidade.

- **Sincronização de operações com precedência** - Neste caso os veículos apenas podem executar as suas respetivas operações após outro veículo ter executado a uma determinada operação na mesma localização. A título de exemplo, no caso apresentado nesta dissertação é necessário executar a operação de estilhagem antes da operação de carregamento. Pode então ser considerado o tempo entre operações (idealmente nulo) e com recurso a este podem ser levadas em consideração as precedências necessárias entre as operações. A imagem 2.3 é caso representativo deste tipo de sincronização em que a operação 2 só pode ser executada após a operação 1 ter sido executada.

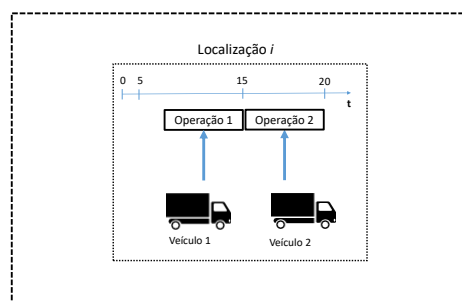


Figura 2.3: Sincronização de operações com precedência

A maioria dos casos que necessitam de sincronização de operações com precedência são problemas do tipo *Dial a Ride* ou *Pick-up and Delivery* com trocas de carga entre veículos. Um tipo de carga pode ser descarregado por um certo veículo num ponto da cadeia de abastecimento destinado ao efeito e ser carregado mais tarde por um veículo diferente. Em

2.1 estão representados alguns artigos sobre sincronização de operações com precedência onde esta é utilizada na resolução de alguns problemas baseados em VRP.

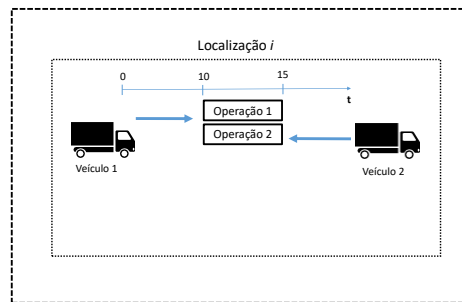
Tabela 2.1: Artigos com sincronização de operações com precedência

Artigo	Variante	Tipos de sincronização	Tipo de variável de decisão
Cortés, Matamala e Contardo [40]	<i>PDP: Passenger transport</i>	Sincronização de operações com precedência, carga fixa	<i>Arc</i>
Wen [41]	<i>PDP with transshipment at a cross-docking centre</i>	Sincronização de operações com precedência, carga fixa	<i>Arc</i>
Bock [42]	<i>Dynamic PDP: Intermodal long-haul transport</i>	Sincronização de operações com precedência, carga fixa	<i>Arc</i>

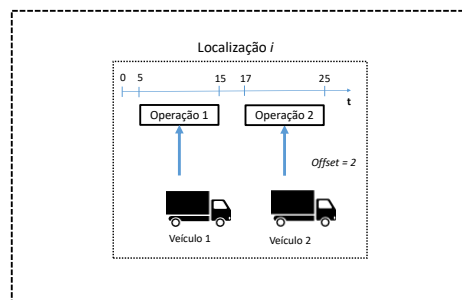
No artigo [40] é explorado o problema de *pick-up and delivery* com transferências; por transferências entende-se, neste caso, a transferência de passageiros entre dois veículos num determinado ponto da rede. A modelação do problema levou a uma sincronização de operações com precedência, em que se garante que a descarga de passageiros acontece antes da carga dos mesmos em todos os pontos da rede. Esta modelação obriga à divisão de um nó em dois outros para cada operação com precedência, o que não se torna viável para a implementação no contexto de biomassa por serem necessárias mais do que uma operação em certas localizações. Já em [41] a sincronização das operações de carga e descarga nos armazéns de *cross-docking* é feita com base nas decisões a tomar no próximo nó da cadeia, sendo mais uma vez inadequada a sua implementação no contexto da biomassa pela existência de várias operações com precedências.

No que diz respeito aos métodos de resolução exatos deste tipo de sincronização, a técnica mais promissora proposta por vários autores [43, 44, 45] é a *Column Generation* em que são determinadas as rotas que os veículos podem tomar assim como as rotas que as encomendas podem tomar. Isto significa que na formulação do problema, para além de existirem variáveis associadas às possíveis rotas que um veículo pode tomar, existem também variáveis associadas às rotas que uma encomenda pode tomar. Se for implementada uma técnica heurística neste tipo de problema, esta tem que se focar em determinar explicitamente as sequências de encomendas possíveis. Os artigos [46, 47] utilizam diferentes métodos na resolução de problemas deste género.

- **Sincronização de operações exatas** - Neste caso os veículos devem executar as suas operações exatamente ao mesmo tempo como representado na imagem 2.4 em que as operações 1 e 2 são executadas no mesmo instante.

Figura 2.4: Sincronização de operações exatas sem *offset*

Por outro lado pode também existir um *offset* fixo para a precedência destas operações, e por ser um aspeto importante para o bom funcionamento das operações não pode ser otimizado. Esta situação é identificada como uma sincronização de operações exatas e está representada na figura 2.5.

Figura 2.5: Sincronização de operações exatas com *offset*

Uma técnica de resolução muito utilizada em problemas de sincronização de operações exatas utiliza uma variável, independente de cada veículo, para o início de execução de uma determinada tarefa ou operação que requer mais do que um veículo no mesmo local [48]. Em formulações que se concentrem nos ramos entre os nós e não nos nós de um problema, esta técnica garante uma sincronização temporal implícita sem que seja necessário adicionar restrições que juntem os diferentes períodos temporais.

2.3.1.3 Sincronização de movimentos

Este tipo de sincronização refere-se ao facto de existirem veículos que não se conseguem movimentar sem a ajuda de outros veículos, destinados ou não para o propósito. Neste casos, os veículos não autónomos requerem um veículo autónomo ou outro veículo não autónomo para se movimentar. Em ambos os casos os movimentos de dois ou mais veículos têm de ser sincronizados no que diz respeito ao período temporal e ao local.

A título de exemplo, no caso apresentado nesta dissertação as estilhadoras podem ser movimentadas entre os pontos de recolha com a ajuda de um veículo destinado para tal.

Assim, o problema de sincronização de movimentos pode-se referir a um certo veículo que, para se movimentar ao longo de um certo arco, um outro veículo tem que se mover ao longo do mesmo arco ao mesmo tempo. Ambos os veículos têm de começar a percorrer o arco ao mesmo tempo, atravessar o arco ao mesmo tempo e chegar ao fim do arco ao mesmo tempo. É necessário ter em conta que esta definição restringe a execução de tarefas ao longo do mesmo arco

Existem dois tipos distintos de sincronização de movimentos.

- **Sincronização de movimentos no depósito** - Neste caso os objetos apenas se podem juntar e separar no depósito central, antes de começar uma rota ou no fim de cada rota.

Nestes problemas existem dois tipos de objetos não autónomos que necessitam de sincronização de movimentos no depósito central, ou seja, ambos os objetos percorrem uma rota completa juntos. Embora na maior parte dos casos os objetos considerados são veículos e condutores, existem várias aplicações para este tipo de sincronização dependendo do que se considera um objeto não autónomo em casos do mundo real. De notar que em problemas VRPTT os camiões de carga são considerados veículos autónomos, pois está implícito que cada veículo tem o seu próprio condutor fixo, em contraste podem ser considerados os camiões de carga e os condutores como objetos não autónomos pois não existe um condutor fixo para cada veículo. A figura 2.6 é um exemplo deste tipo de sincronização, onde neste caso o veículo B necessita do veículo A para se movimentar e, como a sua conexão só é possível no depósito os veículos estão obrigados a manter a mesma rota até voltarem ao depósito.

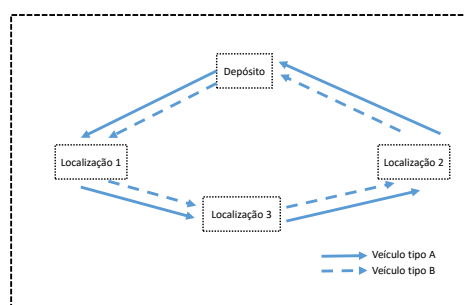


Figura 2.6: Sincronização de movimentos no depósito central

No que diz respeito a abordagens de resolução, algumas soluções determinam rotas abstratas numa primeira fase e só depois cada condutor é associado com cada veículo [49]. Na determinação das rotas na primeira fase do algoritmo é necessário ter em conta algumas limitações para que seja possível obter soluções adequadas nas fases seguintes. Por exemplo, as limitações de condução de cada condutor são consideradas na construção das rotas ao inserir um limite superior na duração de cada rota.

Outro tipo de soluções foca-se no oposto, ou seja, as rotas são determinadas para um conjunto de pares entre veículo e condutor pré-determinados [50]. As soluções iniciais são determinadas ao atribuir tarefas a pares compatíveis. Depois deste passo são determinados pares alternativos para determinadas rotas e então avaliados através de métodos de pesquisa local.

Para este tipo de sincronização os veículos e os condutores só se podem separar ou juntar numa única localização. Isto sugere uma decomposição por tipo de objetos, semelhante à decomposição por estados referida no problema de sincronização de operações no domínio do espacial, tendo em conta os aspetos temporais presentes na sincronização de movimentos.

- **Sincronização de movimentos em rotas** - Em contraste, neste tipo de sincronização os objetos podem-se juntar ou separar durante as suas respetivas rotas, em diferentes localizações. São necessários objetos autónomos, compostos por dois ou mais tipos de objetos elementares autónomos e/ou não autónomos, para realizar certas tarefas. Os objetos elementares podem-se juntar ou separar em várias localizações. Quando comparada com a sincronização de movimentos no depósito central, esta aumenta o espaço de soluções admissíveis do problema e consequentemente a sua complexidade.

A figura 2.7 é um pequeno exemplo deste tipo de sincronização onde os veículos B1 e B2 são idênticos e podem ser utilizados em todas as localizações, contudo necessitam de um veículo A para se movimentarem. A conexão e separação entre veículos pode ser executada em qualquer localização, o que permite aos veículos efetuar rotas mais eficientes.

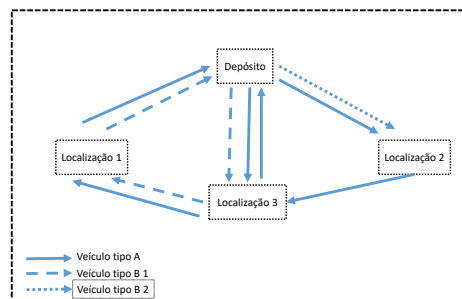


Figura 2.7: Sincronização de movimentos em rotas

Na revisão bibliográfica deste tipo de sincronização a pesquisa efetuada mostra que o número total de publicações sobre sincronização de movimentos em rotas é muito limitado quando comparada com o número de publicações sobre sincronização de movimentos no depósito central.

No que diz respeito a modelos e métodos de resolução baseados em Programação Inteira Mista existem quatro tipos de formulação matemática normalmente utilizada:

- A formulação padrão com três índices X_{ij}^k - Indica se o veículo k utiliza o arco (i, j) .

- A formulação com quatro índices $X_{ij}^{kk'}$ - Indica se os veículos k e k' utilizam o arco (i, j) ao mesmo tempo.
- A formulação com variáveis de mudança X_{hij}^k - Indica se o veículo k utiliza o arco (h, i) imediatamente antes de atravessar o arco (i, j) (isto pode ser interpretado como uma mudança de (h, i) para (i, j)).
- A formulação com variáveis padrão de rotas λ_p^k - Indica se o veículo k percorre a rota p .

As variáveis do tipo $X_{ij}^{kk'}$ garantem a sincronização de movimentos e no modelo matemático são considerados onde e quando os veículos se vão juntar ou separar. Com os outros tipos de variáveis são necessárias restrições adicionais que garantam a conexão entre as variáveis de fluxo e as variáveis temporais nos nós da rede e, essencialmente requerem que se um veículo utiliza um certo arco outro veículo compatível tem que utilizar o mesmo arco na mesma altura.

No que diz respeito ao problema apresentado nesta dissertação, este tipo de sincronização existe, pois para mover as estilhadoras entre os diferentes pontos de recolha é necessária uma sincronização de movimentos entre os veículos autónomos e os veículos não autónomos.

As soluções presentes na literatura não refletem todos os aspetos do problema em questão e é então necessário o desenvolvimento de uma nova solução. Na tabela 2.2 estão representados alguns artigos sobre sincronização de movimentos em rotas que não podem ser utilizados como base para a resolução deste problema.

Tabela 2.2: Artigos com sincronização de movimentos em rotas

Artigo	Variante	Tipos de sincronização	Tipo de variável de decisão
Kim, Koo, e Park [51]	<i>VRP: Staff Scheduling</i>	Sincronização de movimentos em rotas, operações com precedência	<i>Arc</i>
Hollis e Forbes [52]	<i>PDP: Mail Delivery</i>	Sincronização de movimentos em rotas, operações com precedência	<i>Path</i>
Cheung [53]	<i>PDP: Seaport Container</i>	Sincronização de movimentos em rotas, operações exatas, carga fixa	—

Em [51] é feita uma sincronização de movimentos entre os veículos e as diferentes equipas, sendo que cada equipa tem qualificações diferentes e está preparada para executar diferentes tarefas. No problema de recolha e transporte de biomassa apenas são considerados veículos com diferentes características, sendo por isso necessária uma sincronização entre estes veículos, algo que não acontece no artigo apresentado. Tal como no exemplo anterior em [52], apenas é feita uma sincronização de veículos e de equipas e não uma sincronização entre diferentes tipos de veículos. Já em [53] é feita uma sincronização entre três tipos de objetos

diferentes, porém não é feita a distinção entre os diferentes veículos sendo assim impossível uma sincronização dos mesmos.

No que diz respeito a métodos de resolução aproximados, as soluções apresentadas na literatura são muito variadas. Em [51] é desenvolvida uma heurística simples e eficaz para a sincronização de equipas, que executam um certo serviço, com veículos destinados ao transporte dessas equipas. As equipas, os veículos e as próximas tarefas a ser cumpridas são guardadas em três listas, em conjunto com informação relevante sobre aspetos temporais e localizações. Em cada iteração é escolhido um objeto de cada lista utilizando um critério de melhor ajuste. Após cada iteração as listas são atualizadas para refletirem a ultima informação resultante da escolha do veículo da equipa e da localização onde a próxima tarefa tem de ser executada.

2.3.1.4 Sincronização de cargas

A quantidade de capacidade utilizada em cada veículo, quando este está a executar uma tarefa ou uma operação, tem que ser tida em consideração. Por outras palavras, tem que ser garantido que a quantidade certa de carga é recolhida, transportada ou processada.

Para cada nó na rede de abastecimento, com procura negativa, positiva ou nula, a diferença entre o total de carga que foi descarregada num nó por todos os veículos autónomos e a quantidade total de carga descarregada no mesmo nó por todos os veículos não autónomos tem que ser igual a uma procura específica.

Existem três tipos de sincronização de carga.

- **Sincronização de carga fixa** - Neste caso a quantidade de carga que pode ser entregue, recolhida ou transportada está fixada.
- **Sincronização de carga discreta** - Existe um número finito de quantidades de carga disponíveis para transporte ou recolha. Por exemplo, só serem possíveis carregamentos múltiplos de dez de um certo produto.
- **Sincronização de carga contínua** - A quantidade de carga que pode ser recolhida ou transportada apenas está limitada entre zero e um limite superior estabelecido.

Quando num *Vehicle Routing Problem with Multiple Synchronization Constraints* apenas existem sincronizações de tarefas e de carga o problema é então um tipo especial de VRP, é um *Split Delivery VRP* ou um *Pickup and Delivery Problem (PDP)*; ambas as variantes são apresentadas numa secção anterior da dissertação. Nestes problemas é possível que vários veículos visitem o mesmo consumidor, cada um transportando uma parte da procura desejada por esse cliente. Sincronização de carga é necessária nos nós considerados clientes na rede de abastecimento.

2.3.1.5 Sincronização de recursos

A sincronização de recursos é necessária quando a quantidade de recursos disponíveis é limitada e existem vários veículos a competir pelos mesmos recursos. Em qualquer altura do algoritmo a utilização ou consumo total de um certo recurso tem que ser menor ou igual a um limite estabelecido anteriormente.

Este tipo de sincronização foi introduzido por Hemphs e Irnich em 2008, com o nome de *intertour resource constraints*. Em [54] foi desenvolvido um modelo para representar VRP com sincronização de recursos. O modelo é baseado no *framework* unificado desenvolvido por Irnich e utiliza uma representação de rotas gigantes. A ideia passa por considerar uma rota gigante em vez de considerar várias rotas, rota esta que é considerado o caminho mais pequeno restringido pelo recursos disponíveis. Com esta interpretação podem ser usadas em problemas de sincronização de recursos as técnicas de resolução de procura local normalmente utilizadas em VRP sem sincronização de recursos.

Em [55] os autores estudam uma aplicação deste tipo de sincronização na área de planeamento florestal. Neste caso os veículos transportam madeira desde pontos de recolha, situados em florestas, até ao consumidor. Para efetuar o carregamento da madeira para os veículos são necessárias máquinas especiais capazes de carregar um veículo de cada vez, existindo apenas uma máquina por área. Para uma boa otimização deste problema é necessário então considerar a sincronização das máquinas e veículos (recursos) em cada área.

2.3.2 Sumário dos tipos de sincronização

Na tabela 2.3 está representado uma pequena descrição desta classificação para os diferentes tipos de sincronizações presentes em problemas do tipo VRP.

Tabela 2.3: Sumário dos tipos de sincronização

Tipo de sincronização	Descrição
Sincronização de tarefas	Tem objetivo de decidir que veículo é que vai executar a tarefa.
Sincronização de operações	Pode ser dividida em 3 tipos: no domínio espacial onde só é necessário ambos os veículos visitarem a mesma localização, com precedência onde existe uma precedência entre operações e exatas onde as operações são simultâneas ou com offset fixo.
Sincronização de movimentos	Um veículo necessita de outro veículo para se movimentar e subdivide-se em 2 tipos: no depósito , onde só é possível juntar/separar um veículo do seu veículo de transporte no depósito, nas rotas onde a junção/separação é possível em mais localizações
Sincronização de cargas	Ocorre quando se transportam diferentes tipos de cargas e a quantidade em transporte tem um grande impacto na otimização do problema. Pode ser dividida em 3 tipos: carga fixa , onde só é possível transportar uma certa quantidade; carga discreta , onde é possível transportar múltiplos de uma quantidade e carga contínua , que só é limitada pela capacidade do veículo
Sincronização de recursos	Ocorre quando 2 veículos competem pelo mesmo recurso numa área delimitada.

2.4 Home Health Care Routing and Scheduling Problem

A Organização Mundial de Saúde anunciou que a taxa de crescimento de pessoas que necessitam de ajuda e cuidados especiais vai aumentar nas próximas décadas [56]. Esta situação é causada por vários fatores como, por exemplo, a diminuição das taxas de natalidade e o aumento do tempo médio de vida. Para além disso, as famílias estão cada vez mais a viver em localizações dispersas, o que leva a dificuldades no que diz respeito à prestação de cuidados aos familiares idosos mais próximos.

Hoje em dia, a prestação de cuidados é, cada vez mais, transferida para as empresas de cuidados ao domicílio. Estas empresas oferecem uma grande variedade de serviços ao domicílio a pessoas com dificuldades. Para ser possível efetuar uma grande quantidade de serviços, a empresa tem que empregar uma equipa de elementos com diferentes qualificações, incluindo enfermeiros, terapeutas, trabalhadores sociais, fisioterapeutas, entre muitos outros. Alguns tipos de serviço só podem ser efetuados por múltiplos elementos da equipa como, por exemplo, a prestação de assistência a um paciente que necessite de duas ou mais pessoas para ser levantado.

Os elementos da equipa estão normalmente equipados com veículos destinados ao seu próprio transporte, como bicicletas, carros ou motas. Os pacientes requerem um certo tipo de serviço, de preferência dentro de um certo período temporal.

Apesar de relativamente recente o HHCRSP tem sido alvo de alguma investigação. As diferentes formulações do problema variam em quatro aspetos importantes. Assumir equipa de elementos homogénea ou heterogénea, considerar ou não a interdependência de serviços, os objetivos particulares de cada caso no que diz respeito ao planeamento e a metodologia utilizada para a resolução do problema.

Em [57] é considerada uma equipa homogénea sem serviços interdependentes, sendo esta a interpretação mais simples deste tipo de problema, já em [58] também é considerada uma equipa homogénea mas já é considerada a interdependência entre os serviços. Em [59] é considerada uma equipa heterogénea mas não é tida em conta a interdependência entre os serviços. O caso mais próximo da realidade é quando são consideradas uma equipa de elementos heterogéneos e a interdependência entre os serviços [60].

No problema apresentado por Mankowska [61] sobre uma possível interpretação do HHCRSP é considerada uma equipa heterogénea assim como a interdependência entre serviços. Para o efeito são então considerados serviços individuais e serviços duplos.

Um serviço individual é um serviço que consiste numa operação a ser efetuada por apenas um único elemento. Serviços duplos consistem em duas operações a ser efetuadas por dois elementos e podem ser divididas em serviços simultâneos ou serviços com precedência. Um serviço duplo simultâneo é necessário no contexto da saúde se, por exemplo, levantar um paciente requer mais do que um elemento. Serviços duplos com precedência ocorrem quando, por exemplo, certos medicamentos têm de ser administrados horas antes de uma refeição.

Um exemplo para cada tipo de serviço é apresentado na tabela 2.4.

Tabela 2.4: Tipos de serviços e exemplos

Tipo de serviço	Exemplo
Individual	Um elemento da equipa desloca-se a casa de um paciente para administrar medicação.
Duplo simultâneo	Dois elementos da equipa deslocam-se a casa de um paciente para o mover (nos casos em que são necessárias duas pessoas para esta operação)
Duplo com precedência	Dois elementos deslocam-se a casa de um paciente, um após o outro, para administrar medicação e meia hora depois para ajudar o paciente a comer.

Normalmente este tipo de serviços duplos são planeados manualmente, o que leva a uma má qualidade de planeamento e uma eficiência de transporte muito baixa. Para contrariar esta má qualidade de planeamento foi desenvolvido um modelo matemático orientado aos serviços efetuados por um ou dois elementos da equipa.

2.4.1 Formulação matemática

No modelo sugerido por Mankowska é considerado um conjunto de pacientes C , um conjunto de serviços S e uma equipa de membros V . Cada membro de V tem certas qualificações, expressas por uma variável binária que toma o valor 1 se o elemento da equipa tem capacidades para efetuar o serviço S , e toma o valor 0 se não tiver qualificações para efetuar o serviço. Cada paciente C tem requisitos de serviços associados através de uma variável binária que toma o valor 0 se o paciente C não necessita do serviço S e vice versa.

No que diz respeito aos serviços duplos, é considerada uma distância temporal mínima e máxima para que seja possível considerar as precedências necessárias entre os serviços; no caso de estes valores serem nulos, os serviços têm de ser efetuados simultaneamente.

Para além dos parâmetros referidos, no modelo sugerido por Mankowska, são também consideradas as distâncias entre dois nós, assim como o período temporal necessário para efetuar cada serviço. As variáveis consideradas estão representadas na tabela 2.5.

Tabela 2.5: Formulação matemática

Conjuntos e parâmetros	Descrição
C	Conjunto de todos os pacientes.
S	Conjunto de todos os serviços
V	Conjunto de todos os elementos
a_{vs}	1 se o elemento de V está qualificado para efetuar o serviço de S
r_{is}	1 se o paciente de C requer o serviço de S
C^0	Conjunto de todas as localizações; $C^0 = C \cup 0$
C^s	Conjunto de todos os paciente que requerem um serviço individual
C^d	Conjunto de todos os paciente que requerem um serviço duplo $C^d = C^{sim} \cup C^{prec}$
C^{sim}	Conjunto de todos os paciente que requerem um serviço duplo simultâneo
C^{prec}	Conjunto de todos os paciente que requerem um serviço duplo com precedências
δ_i^{min}	Distância temporal mínima entre o começo dos serviços no paciente i
δ_i^{max}	Distância temporal máxima entre o começo dos serviços no paciente i
$[e^i, l^i]$	Janela temporal para o paciente i
d_{ij}	Distancia entre as localizações i e j
p_{is}	Tempo necessário para efetuar o serviço de S no paciente i

As rotas que os elementos da equipa vão efetuar são representadas pelas variáveis de decisão x_{ijvs} que tomam o valor 1 se o elemento v move-se de i para j para efetuar o serviço s , e tomam o valor 0 se o contrário for verdade. São necessárias variáveis de decisão que indiquem quando é que cada serviço começa a ser efetuado, assim como variáveis que avaliem o atraso entre cada serviço. As variáveis consideradas por Mankowska estão representadas na tabela 2.6.

Tabela 2.6: Variáveis de decisão

Variáveis de decisão	Descrição
x_{ijvs}	1 se o elemento de V se move de i para j para efetuar o serviço s
t_{ivs}	Hora de início do serviço s no paciente i fornecida pelo elemento v
z_{is}	Atraso do serviço s no paciente i

2.5 Enquadramento do problema no referencial de sincronização

O problema apresentado nesta dissertação requer vários tipos de sincronização. Como existem veículos autónomos específicos para transportar as estilhadoras, consideradas veículos não autónomos, existe uma sincronização de movimentos. Esta sincronização pode ser efetuada em qualquer dos pontos de recolha disponíveis na rede, existe então uma sincronização de movimentos nas rotas. Pode também ser considerada uma sincronização de operações, na medida em que é necessário descarregar as estilhadoras e estilhar os resíduos florestais antes de os carregar nos camiões de transporte, e uma sincronização de cargas atendendo ao facto que deve ser tomada uma decisão sobre a quantidade de matéria prima a ser transportada e estilhada, tendo em conta os valores energéticos da mesma.

A abordagem mais importante para o foco desta dissertação passa por considerar todas as operações como serviços, serviços estes que podem ser executados por certos veículos e necessitam, ou não, de precedências e sincronização entre eles. São também considerados os nós em que estes serviços são necessários assim como os serviços que têm de ser realizados antes de determinados serviços ou em simultâneo. Esta abordagem está aplicada ao problema *Home Health Care Routing and Scheduling Problem* (HHCRSP) e o objetivo desta dissertação passa pela modelação e implementação de uma rede de recolha e transporte de biomassa com base na formulação de Mankowska para o HHCRSP.

Com base na formulação apresentada anteriormente, podem ser considerados os três conjuntos principais apresentados na tabela 2.7, assim como o seu paralelismo com o HHCRSP.

Tabela 2.7: Comparação de conjuntos

Conjuntos	Descrição no HHCRSP	Descrição no contexto de biomassa
C	Conjunto de todos os pacientes	Conjunto de todas as localizações (fornecedores, armazéns, depósito central e consumidores)
S	Conjunto de todos os serviços	Conjunto de todas as operações (serviços) no contexto de biomassa
V	Conjunto de todos os elementos	Conjunto de todos os veículos

Na tabela 2.8 está representado um exemplo de cada tipo de serviço no contexto de transporte e recolha de biomassa.

Tabela 2.8: Tipos de serviços e exemplos no contexto de biomassa

Tipo de serviço	Exemplo
Individual	Carregar a estilhadora no veículo destinado ao seu transporte.
Duplo simultâneo	Operação de estilhagem (serviço 1) tem de acontecer em simultâneo com a operação de carregamento de resíduos (serviço 2)
Duplo com precedência	É necessário descarregar a estilhadora (serviço 1) antes de estilhar os resíduos (serviço 2)

Capítulo 3

Um novo modelo para sincronização em cadeias de abastecimento

Neste capítulo é explorada a nova abordagem ao problema de sincronização de biomassa, baseada na formulação apresentada anteriormente por Mankowska para o HHCRSP. Neste capítulo da dissertação é apresentado o modelo base [62] em conjunto com pequenas alterações a conjuntos e restrições na secção 3.1 que já são considerados como uma contribuição para a dissertação, pois sem elas o modelo não seria viável. A partir da secção 3.2 todo o conteúdo apresentado são contribuições originais para o problema apresentado. O conteúdo apresentado a partir da secção 3.2 é conteúdo original da dissertação e aborda as questões de sincronização de movimentos nas rotas que não são tidos em consideração na abordagem original.

3.1 Modelo matemático

Tal como referido anteriormente, a formulação matemática base neste capítulo é inspirada no HHCRSP de Mankowska estudada pelo INESC TEC no âmbito do projeto Easy Flow iniciado por Ricardo Soares [62].

3.1.1 Entidades do problema

São considerados três conjuntos de objetos principais distintos relevantes para o problema:

- **Conjunto C** - Enumera todas as localizações relevantes para o problema. O conjunto é composto por pontos de recolha de resíduos florestais, terminais destinados ao armazenamento dos resíduos e os consumidores dos mesmos.

Sendo esta formulação orientada aos serviços o conjunto é então dividido no subconjunto C^d , em que estão todas as localizações que necessitam de mais que um serviço sincronizado, e no subconjunto C^s , em que estão representadas todas as localizações que apenas necessitam de um tipo de serviço (sem sincronização necessária). Para além dos subconjuntos definidos também é considerado C^0 que consiste no conjunto C com a adição do depósito

central onde começam e acabam todas as rotas. A tabela 3.1 representa uma pequena descrição das localizações existentes no problema.

Tabela 3.1: Localizações existentes

Localizações	Descrição
Ponto de recolha	Onde são recolhidos e estilhados os resíduos florestais
Armazém	Onde são armazenados os resíduos já estilhados
Fábrica	Para é transportada a biomassa para consumo
Depósito central	Localização inicial e final comum a todos os veículos

- **Conjunto V** - Enumera todos os tipos de veículos disponíveis para serem utilizados. O conjunto contém os camiões de transporte do tipo A, que transportam as máquinas de estilhagem entre os diferentes pontos de recolha, camiões do tipo B, que transportam os resíduos já estilhados desde os pontos de recolha até um armazém ou diretamente até ao consumidor e por último o conjunto inclui as máquinas destinadas ao estilhamento dos resíduos que, apesar de serem veículos não autónomos e não se conseguirem mover autonomamente entre os pontos de recolha, são considerados neste modelo para que seja possível uma sincronização de movimentos.

Tendo em conta os diferentes tipos de veículos, estes são diferenciados apenas pelo tipo de serviços que conseguem executar, sendo então necessário um conjunto de serviços. A tabela 3.2 representa uma pequena descrição da função de cada veículo presente no problema.

Tabela 3.2: Veículos existentes

Veículos	Função
Tipo A	Encarregue do transporte de estilhadoras
Tipo B	Encarregue do transporte de biomassa
Estilhadora	Encarregue do estilhamento de resíduos

- **Conjunto S** - Enumera todos os serviços que são considerados inicialmente no modelo. Este é o conjunto que diferencia esta abordagem de todas as outras e requer uma análise atenta aos serviços considerados, assim como as condições e precedências necessárias para a execução de cada serviço. A tabela 3.3 sumariza todos os serviços considerados no modelo, assim como as suas condições e precedências.

Tabela 3.3: Serviços, respetiva descrição e condições

S	Descrição do serviço (A)	Veículo (B)	Localização (C)	Serviço de precedência do veículo (D)	Serviços de sincronização na localização (E)
1	Carregamento da estilhadora	Tipo A	Depot	Nenhum	Nenhum
			Pontos de recolha	2	
2	Descarregamento da estilhadora	Tipo A	Ponto de recolha	1	Nenhum
3	Estilhar resíduos	Estilhadora	Ponto de recolha	Nenhum, 3	2 (precedência) 4 (simultâneo)
4	Carregamento de resíduos	Tipo B	Ponto de recolha	Nenhum	3 (simultâneo)
			Armazém	5	Nenhum
5	Descarregamento resíduos	Tipo B	Armazém	4	Nenhum
			Fábricas		
6	Voltar para o depósito	Estilhadora	Depósito	3	Nenhum
		Tipo A		1, 2	
		Tipo B		5	

Com os dados da tabela dos serviços 3.3 é necessário criar parâmetros para que seja possível uma correta modelação do problema. A coluna A é apenas uma pequena descrição de cada serviço, já os dados da coluna B representam que veículos podem executar cada serviço, os dados referentes a esta coluna serão representados por uma variável binária a_{vs} . Da mesma forma, os dados da coluna C serão representados pela variável binária r_{is} (descritos na tabela 3.5) e dizem respeito aos serviços que são necessários em cada localização.

Contudo, a informação das colunas A, B e C não é suficiente para descrever o problema de forma apropriada pois, existem serviços que só podem ser executados se outro serviço for executado no momento imediatamente antes pelo mesmo veículo. Com os dados da coluna D são criados dois subconjuntos, o subconjunto G_v representa todos os pares de serviços ordenados (s_1, s_2) em que é necessário executar s_1 antes de executar s_2 . Os serviços que não necessitam desta precedência com outros serviços são incluídos no subconjunto G_v^* . Os veículos devem iniciar as suas operações a partir de um dos serviços contidos em G_v^* .

Da mesma forma é necessário considerar a sincronização destes serviços nas localizações, isto é, existem localizações que requerem que outro tipo de serviço seja executado imediatamente antes ou durante a execução de um serviço distinto. Esta informação está representada na coluna E da tabela 3.3 e é representada no modelo pelo conjunto de pares de serviços ordenados H_i . Esta sincronização é mais evidente nos pontos de recolha onde são necessários dois tipos de veículos a executar diferentes serviços.

3.1.2 Formulação matemática

Nesta secção da dissertação vão ser apresentados todos os conjuntos, parâmetros, variáveis de decisão e restrições que são consideradas no modelo, assim como a sua função objetivo. A explicação para cada um dos pontos está nas secções seguintes.

Conjuntos

Tabela 3.4: Conjuntos aplicada ao contexto do problema

Conjunto	Descrição
C^s	Conjunto das localizações que necessitam de 1 serviço
C^d	Conjunto das localizações que necessitam de 2 serviços
C	Conjunto de todas as localizações $C = C^s \cup C^d$
C^0	Conjunto de localizações, incluindo o depósito central $C^0 = C \cup 0$
V	Conjunto de todos os veículos
V^e	Conjunto de todas as estilhadoras
V^b	Conjunto de todos os veículos do tipo B
V^a	Conjunto de todos os veículos do tipo A
S	Conjunto de todos os serviços, $S = 1, 2, \dots, S $
G_v^*	Conjunto de serviços iniciais por cada veículo
G_v	Conjunto de pares de serviços ordenados (s_1, s_2) que requerem que o veículo $v \in V$ tenha executado s_1 antes de executar s_2 , $(s_1, s_2) \in S^2$, $s_1 \neq s_2$
H_i	Conjunto de pares de serviços ordenados (s_1, s_2) que requerem que tenha sido executado s_1 antes de s_2 na localização $i \in C^d$, $(s_1, s_2) \in S^2$, $s_1 \neq s_2$
H_i^*	Conjunto de pares de serviços ordenados (s_1, s_2) que requerem que s_1 seja executado ao mesmo tempo que s_2 na localização $i \in C^d$, $(s_1, s_2) \in S^2$, $s_1 \neq s_2$

Parâmetros

Tabela 3.5: Parâmetros adicionais

Parâmetro	Descrição
a_{vs}	Variável binária com o valor 1 se $v \in V$ pode executar o serviço $s \in S$, 0 se não
r_{is}	Variável binária com o valor 1 se $s \in S$ é necessário em $i \in C^0$, 0 se não
d_{ij}	Distância temporal entre as localizações $i, j \in C^0$
p_{is}	Duração do serviço $s \in S$ na localização $i \in C^0$
p_{iks}	Duração do serviço $s \in S$ a executar pela estilhadora $k \in V^e$ na localização $i \in C^0$
c_v, c_v^F	Custo por unidade de tempo e custo fixo por veículo $v \in V$
T_{start}, T_{end}	Limites mínimo e máximo da linha temporal

Variáveis de decisão

Tabela 3.6: Variáveis de decisão e descrição

Variável	Descrição
x_{ijvs}	Variável binária com o valor 1 se $v \in V$ vai de $i \in C^0$ para $j \in C^0$ para executar $s \in S$, 0 caso contrário
t_{ivs}	Tempo de começo de $s \in S$ na localização $i \in C^0$ por $v \in V$

Função objetivo

$$\min F = \sum_{v \in V} \sum_{s \in S} c_v * t_{0vs} + \sum_{i \in C^0} \sum_{v \in V} \sum_{s \in S} c_v^F * x_{i0vs} \quad (3.1)$$

Sujeita a:

$$\sum_{i \in C^0} \sum_{s \in G_v^*} x_{0ivs} = \sum_{i \in C^0} \sum_{s \in S} x_{i0vs} \quad \forall v \in V \quad (1)$$

$$\sum_{i \in C^0} \sum_{s \in S} x_{ijvs} = \sum_{i \in C^0} \sum_{s \in S} x_{jivs} \quad \forall j \in C^0, v \in V \quad (2)$$

$$\sum_{v \in V} \sum_{j \in C^0} a_{vs} * x_{jivs} = r_{is} \quad \forall i \in C^0, s \in S \quad (3)$$

$$\sum_{j \in C} a_{vs_6} * x_{j0vs_6} = r_{is_6} \quad v \in V \quad (3.i)$$

$$T_{start} + d_{0j} \leq t_{jvs} + M_1 * (1 - x_{0jvs}) \quad \forall j \in C^0, v \in V, s \in G_v^* \quad (4)$$

$$t_{ivs_1} + p_{is_1} + d_{ij} \leq t_{jvs_2} + M_1 * (1 - x_{ijvs_2}) \quad \forall i \in C, j \in C^0, v \in V, (s_1, s_2) \in G_v \quad (5)$$

$$t_{iv_2s_2} - t_{iv_1s_1} \geq p_{is_1} - M_2 * (2 - \sum_{j \in C^0} x_{jiv_1s_1} - \sum_{j \in C^0} x_{jiv_2s_2}) \quad \forall i \in C^d, v_1, v_2 \in V, (s_1, s_2) \in H_i \quad (6)$$

$$x_{ijvs} \in \{0, a_{vs} * r_{js}\} \quad \forall i, j \in C^0, v \in V, s \in S \quad (7)$$

$$a_{vs} * r_{is} * T_{start} \leq t_{ivs} \leq a_{vs} * r_{is} * T_{end} \quad \forall i \in C^0, v \in V, s \in S \quad (8)$$

$$t_{ivs} \leq M_4 * \sum_{j \in C^0} x_{jivs} \quad \forall i \in C^0, \forall v \in V, \forall s \in S \quad (9)$$

3.1.3 Variáveis de decisão e função objetivo

Para ser possível modelar as decisões a ser tomadas neste problema é necessário recorrer a variáveis de decisão. Estas variáveis indicam se um veículo executa uma certa rota para fornecer um serviço e o momento em que se inicia cada serviço nas localizações. As variáveis utilizadas estão representadas na tabela 3.6.

Com recurso às variáveis de decisão apresentadas na tabela 3.6 e aos parâmetros apresentados na tabela 3.5 é possível construir uma equação que funciona como objetivo do problema. Neste caso a função objetivo pretende minimizar os custos relacionados com o transporte de resíduos. Na função objetivo apresentada em 3.1 são considerados os custos por período temporal assim como os custos fixos associados à utilização de cada veículo.

3.1.4 Restrições

Num problema de otimização combinatória é necessário restringir as soluções que podem ser obtidas através de restrições. As restrições são associadas a situações que não podem acontecer no modelo.

No modelo apresentado são consideradas as seguintes restrições:

A restrição 1 indica que todos os veículos considerados no conjunto V que começam uma rota no depósito central têm que a acabar no mesmo. A restrição 2 garante a continuidade do fluxo, isto é, todos os veículos que entram num nó da cadeia são obrigados a sair desse mesmo nó. Já a restrição 3 garante que é atribuído um veículo para efetuar cada serviço em cada localização. No entanto é necessário garantir que ao nível dos serviços todos os veículos efetuam o serviço S_6 na mesma localização (depósito central), a restrição $3.i$ é uma adaptação da anterior e garante isso mesmo.

As restrições 4 e 5 dizem respeito à sincronização da precedência de serviços dos veículos representados pela coluna D da tabela 3.3. A restrição 4 garante que o primeiro serviço de cada veículo apenas é executado depois do veículo percorrer a distância temporal entre o depósito central e o nó de destino. A restrição 5 obriga a que, no par de serviços ordenados (s_1, s_2) , seja executado o serviço s_1 antes do serviço s_2 pelo veículo v .

A restrição 6 diz respeito às precedências de serviços em localizações apresentadas na coluna E da tabela 3.3. A restrição garante então que, tendo em conta o par de serviços ordenados (s_1, s_2) , na localização i o serviço s_1 é executado antes do serviço s_2 .

Em 7 e em 8 são definidos os domínios das variáveis de decisão. Em 7 é definido que apenas os veículos que têm as capacidades necessárias para executar o serviço s se podem deslocar à localização j para o executar. Já em 8 é garantido que o início do serviço s na localização i não ultrapassa os limites inferior ou superior definidos.

Por fim é necessário garantir que, quando uma rota não é utilizada para executar um serviço, o seu respetivo tempo de início tem que ser nulo, evitando assim soluções inválidas causadas pelas restrições de sincronização. Para garantir este aspeto é utilizada a restrição 9.

De notar que as restrições apresentadas são baseadas na formulação de Ricardo Soares e contêm pequenas alterações que tornam o modelo viável [62]. Com a exceção das restrições 1, 2 e 4 serem apresentadas no seu estado original.

3.2 Extensão do modelo

O modelo matemático apresentado na secção anterior não se encontra completo, pois não são considerados dois tipos de sincronização fundamentais para o problema, a sincronização de movimentos e a sincronização de operações em simultâneo.

A sincronização de movimentos visa garantir que os veículos não autónomos e os veículos do tipo A, destinados ao seu transporte, percorrem o mesmo arco (i, j) ao mesmo tempo, garantindo assim que para transportar os veículos não autónomos são necessários os veículos do tipo A.

A sincronização de operações em simultâneo é utilizada quando, por exemplo, é necessário garantir que os serviços de estilhagem e carregamento dos camiões do tipo B acontecem ao mesmo tempo.

3.2.1 Sincronização de operações em simultâneo

A sincronização de operações em simultâneo é um caso especial de sincronização de operações com precedência, em que a precedência considerada tem um valor nulo, obrigando assim a que ambos os serviços comecem no mesmo instante.

Este tipo de sincronização pode ser alcançada através de uma variação da restrição 6, nesta variação o parâmetro p_{is} toma um valor nulo, obrigando assim a que os serviços comecem no mesmo instante.

No caso apresentado, este tipo de sincronização é necessário para vários pares de serviços, pelo que se pode considerar um conjunto H_i^* que contém todos os pares de serviços ordenados (s_1, s_2) em que este tipo de sincronização é necessária na localização i .

Tabela 3.7: Novo conjunto considerado

Conjunto	Descrição
H_i^*	Conjunto de pares de serviços ordenados (s_1, s_2) que requerem que s_1 seja executado ao mesmo tempo que s_2 na localização $i \in C^d$, $(s_1, s_2) \in S^2, s_1 \neq s_2$

Tabela 3.8: Novos parâmetros considerados

Parâmetro	Descrição
p_{iks}	Duração do serviço $s \in S$ a executar pela estilhadora $k \in V^*$ na localização $i \in C$

Com a divisão sugerida a restrição 6 e a sua variação 6.i tomam então uma nova forma. De notar que neste caso apenas existe um conjunto de pares que se inserem na restrição 6 (par ordenado $[s_3, s_1]$), desta forma o parâmetro p_{is} pode ser alterado para o parâmetro p_{iks} apresentado na tabela 3.8 tendo em conta que o novo parâmetro é específico das estilhadoras.

$$t_{iv_2s_2} - t_{iv_1s_1} \geq p_{iks_1} - M_2 * (2 - \sum_{j \in C^0} x_{jiv_1s_1} - \sum_{j \in C^0} x_{jiv_2s_2}) \quad \forall i \in C^d, v_1, v_2 \in V, k \in V^e, (s_1, s_2) \in H_i \quad (6)$$

$$t_{iv_2s_2} - t_{iv_1s_1} \geq -M_2 * (2 - \sum_{j \in C^0} x_{jiv_1s_1} - \sum_{j \in C^0} x_{jiv_2s_2}) \quad \forall i \in C^d, v_1, v_2 \in V, (s_1, s_2) \in H_i^* \quad (6.i)$$

3.2.2 Sincronização de movimentos

A abordagem proposta nesta dissertação para garantir a sincronização de movimentos passa pela adição de dois novos serviços apresentados na tabela 3.9.

Tabela 3.9: Dois novos serviços

S	Descrição do serviço (A)	Veículo (B)	Localização (C)
7	Transporte e descarregamento da estilhadora	Tipo B	Arco (i,j)
8	Estilhadora em transporte	Estilhadora	Arco (i,j)

O serviço s_7 diz respeito ao transporte e descarregamento da estilhadora por um camião do tipo A e não tem uma localização definida mas sim um arco entre o par de localizações (i, j) . O serviço s_8 tal como o serviço s_7 diz respeito ao transporte da estilhadora mas na perspetiva da própria estilhadora, tal como em s_7 , s_8 não tem uma localização definida mas sim o arco entre o par de localizações (i, j) . O arco (i, j) pode ser considerado, em ambos os casos, como o nó j . De notar que com a inserção do serviço s_7 o serviço s_2 deixa de existir.

Para garantir a sincronização de movimentos é então necessário que os serviços s_7 e s_8 comecem ao mesmo tempo na mesma localização. Ao inserir estes serviços no conjunto H_i^* esta sincronização fica garantida através de uma adaptação da restrição 6.i. A adaptação efetuada em 6.ii é necessária para garantir que os serviços considerados de "movimento" são efetuados por ambos os veículos no mesmo nó.

$$t_{iv_2s_2} - t_{iv_1s_1} \geq -M_2 * (2 - \sum_{j \in C} x_{jiv_1s_1} + x_{jiv_2s_2}) \quad \forall i \in C^d, v_1, v_2 \in V, (s_1, s_2) \in H_i^* \quad (6.ii)$$

A restrição 6.ii garante então que o arco (i, j) é percorrido pelo veículo não autónomo e pelo veículo autónomo mesmo tempo.

Com a inclusão dos serviços s_7 e s_8 já não é necessária a utilização do serviço s_6 (Voltar ao depósito) por parte dos veículos do tipo A e pelas estilhadoras, ao considerar que o movimento descrito por estes serviços pode ser representativo do serviço s_6 para diferentes veículos. Com este pressuposto são necessárias mais variações da restrição 3.i. As restrições 3.ii e 3.iii têm o mesmo objetivo que 3.i já descrito anteriormente.

$$\sum_{j \in C} a_{vs_7} * x_{j0vs_7} = r_{is_7} \quad v \in V \quad (3.ii)$$

$$\sum_{j \in C} a_{vs_8} * x_{j0vs_8} = r_{is_8} \quad v \in V \quad (3.iii)$$

A inclusão dos serviços s_7 e s_8 altera então a tabela dos serviços apresentada anteriormente na medida em que, ao considerar os serviços intermédios as precedências entre os serviços vão ser alteradas. A nova tabela já não considera o serviço s_2 , visto este estar incluído em s_7 . A tabela apresentada em 3.10 apresenta todas as alterações de precedências mencionadas, assim como as alterações aos serviços considerados de "movimento" (s_6, s_7, s_8).

Tabela 3.10: Nova tabela de Serviços, respetiva descrição e condições

Serviço	Descrição do serviço (A)	Veículo (B)	Localização (C)	Serviço de precedência do veículo (D)	Serviços de sincronização na localização (E)
1	Carregamento da estilhadora	Tipo A	Depot	Nenhum	3(precedência)
			Ponto de recolha	7	
3	Estilhar resíduos	Estilhadora	Ponto de recolha	8	Nenhum
4	Carregamento de resíduos	Tipo B	Ponto de recolha	Nenhum, 5	3(simultâneo)
			Armazém		Nenhum
5	Descarregamento de resíduos	Tipo B	Armazém	4	Nenhum
			Fábricas		
6	Voltar para o depósito	Tipo B	Depósito	5	Nenhum
7	Transporte e descarregamento da estilhadora	Tipo A	Depósito	1	8(simultâneo)
			Ponto de recolha	Nenhum, 1	
8	Estilhadora em transporte	Estilhadora	Depósito	3	7(simultâneo)
			Ponto de recolha	Nenhum, 3	

3.2.3 Considerações adicionais

O modelo apresentado ainda necessita de alguns ajustes para que seja possível obter soluções válidas. É necessário existir uma exclusividade entre alguns serviços para as diferentes localizações, garantindo assim que as soluções geradas não contêm fluxos considerados errados. Para garantir esta exclusividade foi criado um serviço s_9 igual ao serviço s_4 que apenas difere na sua localização, sendo s_9 executado apenas nos armazéns. Em relação aos serviços considerados de movimento foram criados dois novos serviços s_{10} e s_{11} , iguais a s_7 e s_8 respetivamente mas exclusivos para o depósito central. Os novos serviços adicionados estão representados na tabela 3.11.

Tabela 3.11: Novos serviços de exclusividade

Serviço	Descrição do serviço (A)	Veículo (B)	Localização (C)	Serviço de precedência do veículo (D)	Serviços de sincronização na localização (E)
9	Carregamento de resíduos	Tipo B	Armazém	5	Nenhum
10	Transporte e descarregamento da estilhadora	Tipo A	Depósito	1	11(simultâneo)
11	Estilhadora em transporte	Estilhadora	Depósito	3	10(simultâneo)

Com a adição dos novos serviços, as restrições 3.ii e 3.iii passam a considerar os serviços s_{10} e s_{11} em vez de s_7 e s_8 respetivamente.

$$\sum_{j \in C} a_{vs_{10}} * x_{j0vs_{10}} = r_{0s_{10}} \quad v \in V \quad (3.ii)$$

$$\sum_{j \in C} a_{vs_{11}} * x_{j0vs_{11}} = r_{0s_{11}} \quad v \in V \quad (3.iii)$$

Para além do aspeto referido, é necessário garantir que os serviços s_3 e s_4 têm a mesma duração, tendo em conta que a duração do serviço s_3 depende da produtividade da estilhadora e da quantidade de resíduos a estilhar. Os dados referentes à estilhadora são obtidos previamente à resolução deste problema operacional. A produtividade da estilhadora é conhecida pelas características das mesmas, enquanto a quantidade a estilhar em cada ponto de recolha é decidida num nível tático. Para garantir que os serviços têm a mesma duração é necessário criar um subconjunto V^e em que são colocadas todas as estilhadoras e um subconjunto V^b em que são colocados todos os camiões do tipo B. Estes novos conjuntos estão representados na tabela 3.12.

Tabela 3.12: Novos conjuntos considerados

Conjunto	Descrição
V^e	Conjunto de todas as estilhadoras
V^b	Conjunto de todos os camiões do tipo B

Com a adição destes conjuntos é então possível diferenciar as estilhadoras e assim desenvolver uma restrição que garante o próximo serviço a ser executado por um camião do tipo B depois de s_4 (s_5) só pode ser iniciado depois da conclusão de s_3 por uma estilhadora. A restrição 10 garante

esta sincronização adicional.

$$t_{ivs_4} + \sum_{z \in C} (x_{zivs_4} * p_{iks_3}) \leq t_{jvs_5} + Tend * (2 - x_{ijvs_5} - \sum_{z \in C} x_{ziks_3}) \quad \forall v \in V^b, \forall i \in C, \forall j \in C, \forall k \in V^e \quad (10)$$

A nova tabela completa com todos os serviços considerados no modelo está representada na tabela 3.13

Tabela 3.13: Tabela final com todos os serviços

Serviço	Descrição do serviço (A)	Veículo (B)	Localização (C)	Serviço de precedência do veículo (D)	Serviços de sincronização na localização (E)
1	Carregamento da estilhadora	Tipo A	Depot Ponto de recolha	Nenhum 7	3(precedência)
3	Estilhar resíduos	Estilhadora	Ponto de recolha	8	Nenhum
4	Carregamento de resíduos	Tipo B	Ponto de recolha Armazém	Nenhum, 5	3(simultâneo) Nenhum
5	Descarregamento de resíduos	Tipo B	Armazém Fábricas	4	Nenhum
6	Voltar para o depósito	Tipo B	Depósito	5	Nenhum
7	Transporte e descarregamento da estilhadora	Tipo A	Ponto de recolha	Nenhum, 1	8(simultâneo)
8	Estilhadora em transporte	Estilhadora	Ponto de recolha	Nenhum, 3	7(simultâneo)
9	Carregamento de resíduos	Tipo B	Armazém	5	Nenhum
10	Transporte e descarregamento da estilhadora	Tipo A	Depósito	1	11(simultâneo)
11	Estilhadora em transporte	Estilhadora	Depósito	3	10(simultâneo)

3.3 Caso de estudo

Para implementar e testar o modelo foi necessário criar uma instância simples para avaliar as soluções obtidas e validar o modelo desenvolvido. O caso apresentado foi inspirado numa empresa Finlandesa de abastecimento de biomassa a centrais de produção de energia térmica. A instância considerada envolve três pontos de recolha, dois terminais destinados ao armazenamento de resíduos florestais, três fábricas onde são processados os resíduos florestais e convertidos em energia e o depósito central onde começam e acabam todas as rotas. No que diz respeito aos

veículos, nesta instância são considerados dois veículos do tipo A, dois veículos do tipo B e dois estilhadoras.

A imagem 3.1 é uma simples representação da instância considerada. Na imagem estão também representadas um pequeno número das rotas possíveis e consideradas na resolução do modelo.

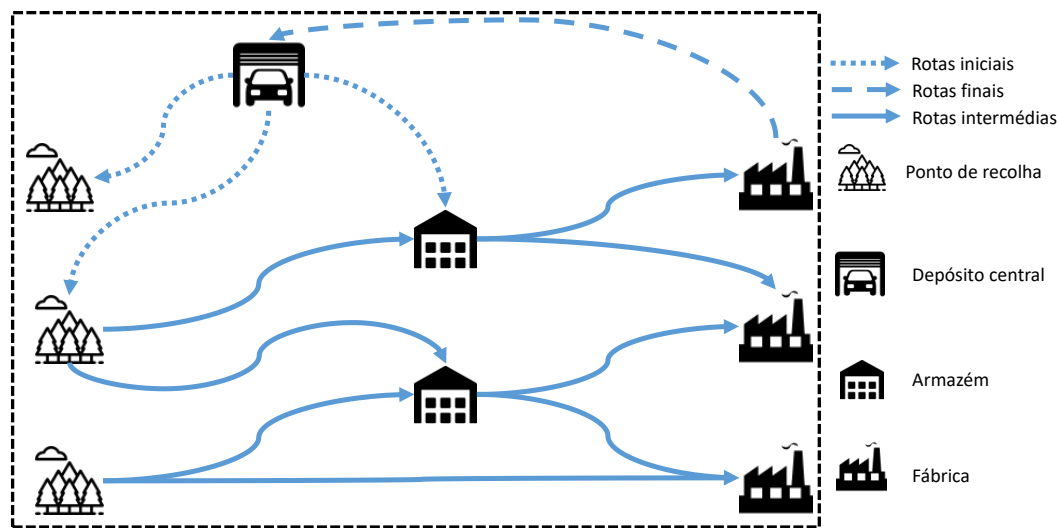


Figura 3.1: Instância considerada

Os três pontos de recolha considerados podem ser diferenciados pela quantidade de resíduos florestais disponíveis e estão representados na tabela 3.14.

Tabela 3.14: Quantidades disponíveis nos pontos de recolha

Localização	Quantidade disponível (m^3)
Ponto de recolha 1	480
Ponto de recolha 2	600
Ponto de recolha 3	370

Os dois armazéns considerados na instância estão representados na tabela 3.15 e podem ser diferenciados pela sua capacidade máxima e pelo inventário inicial.

Tabela 3.15: Capacidades máximas de cada armazém

Localização	Capacidade máxima (m^3)
Armazém 1	300
Armazém 2	300

As fábricas consideradas diferenciam-se pela sua procura e capacidade máxima e estão representadas na tabela 3.16.

Tabela 3.16: Capacidades e procura das fábricas

Localização	Capacidade máxima (m^3)	Procura (m^3)
Fábrica 1	350	195
Fábrica 2	350	195
Fábrica 3	350	195

No que diz respeito aos veículos do tipo A e do tipo B, para além de executarem diferentes serviços, distinguem-se pelo seu custo de transporte por unidade temporal, pelo custo fixo de utilização de cada veículo e pela sua capacidade máxima. Esta distinção está representada na tabela 3.17.

Tabela 3.17: Diferentes custos para os diferentes veículos

Veículo	Custo fixo (€)	Custo de transporte (€/min)	Capacidade máxima (toneladas)
Tipo A 1	7	7	28
Tipo A 2	25	10	48
Tipo B 1	7	7	28
Tipo B 2	25	10	48

Ao contrário dos veículos de tipo A e tipo B, as estilhadoras distinguem-se pela sua produtividade, que é diferente em cada estilhadora e varia entre os diferentes pontos de recolha. Os dados referentes às estilhadoras estão representados na tabela 3.18

Tabela 3.18: Produtividade das estilhadoras nos diferentes pontos de recolha

Veículo	Produtividade no ponto 1 (m^3/h)	Produtividade no ponto 2 (m^3/h)	Produtividade no ponto 3 (m^3/h)
Estilhadora 1	50	45	60
Estilhadora 2	50	45	60

3.3.1 Geração da rede

O modelo apresentado nesta dissertação é um modelo do problema ao nível operacional, pelo que existem vários *inputs* que advêm do nível tático do problema. Na instância apresentada as rotas dos camiões do tipo B já estão definidas previamente, assim como as quantidades a transportar por cada camião. Este tipo de decisões advêm de um nível tático pois num nível superior de decisão são considerados fatores como a qualidade e humidade da matéria-prima, assim como os preços variáveis de cada fábrica. Desta forma é necessário um cuidado especial na geração e pré-processamento da rede. Na tabela 3.19 estão representadas as rotas já decididas num nível tático presentes neste problema.

Numa primeira fase é necessário dividir os nós existentes na rede em nós falsos, isto é, para o bom funcionamento do modelo é necessário um nó exclusivo para cada rota pré-determinada

Tabela 3.19: Rotas previamente definidas

Localização	Destino previamente definido
Ponto de recolha 1	Armazém 2
Ponto de recolha 2	Fábrica 1
Ponto de recolha 3	Fábrica 2
Armazém 1	Fábrica 3
Armazém 2	Fábrica 1

de B, assim como é necessário um nó exclusivo para o transporte das estilhadoras quando são consideradas os pontos de recolha.

Na imagem 3.2 está representada uma divisão simples de um ponto de recolha nos seus nós falsos, assim como os serviços associados a cada nó falso. De notar que o número de nós falsos é determinado pelo número de rotas dos camiões do tipo B já determinadas a um nível tático, isto é, neste caso existem duas rotas para o ponto de recolha pelo que existem dois nó falso destinado ao estilhamento e carregamento de resíduos. De notar que o serviço s_1 só existe no nó considerado como o último do ponto de recolha, desta forma é garantido que todos os resíduos no ponto de recolha são estilhados antes da estilhadora se mover para outro ponto da rede.

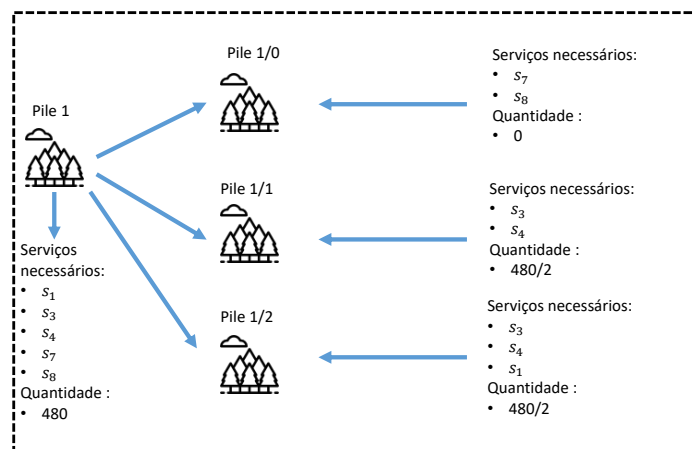


Figura 3.2: Exemplo de divisão de um ponto de recolha em nós falsos

Na imagem 3.3 está um pequeno exemplo de divisão dos armazéns nos seus nós falsos, também determinados pelas rotas dos camiões do tipo B. Neste caso, apesar de representarem a mesma localização, cada nó falso tem um propósito diferente, existem um nó para receber resíduos (s_5), representado por Ter/D1 (*Demand*) e um nó de onde são retirados resíduos (s_9), representado por Ter/S1 (*Supply*). No caso apresentado apenas existem duas rotas decididas para os camiões do tipo B, uma das rotas tem como origem um ponto de recolha logo é um nó falso do tipo *Demand* enquanto a outra rota tem como destino uma fábrica pelo que o nó falso associado é do tipo *Supply*.

De notar que o número de nós falsos está dependente das rotas definidas previamente num nível tático.

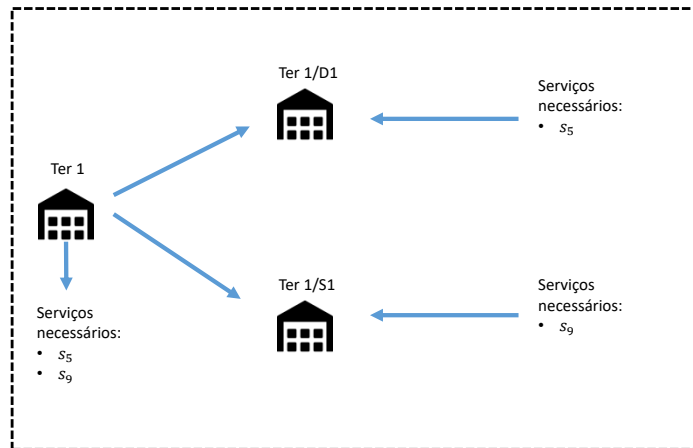


Figura 3.3: Exemplo de divisão de um armazém em nós falsos

Já os nós falsos das fábricas são todos iguais, isto é, têm exatamente o mesmo propósito e os mesmo serviços associados, o seu número depende mais uma vez das rotas dos camiões do tipo B que foram pré-determinadas. A imagem 3.4 é uma pequena representação dos nós falsos das fábricas, onde estão representadas duas rotas para a fábrica, de notar que no caso das fábricas o ponto de origem da rota não influencia o número nem a forma dos nós falsos.

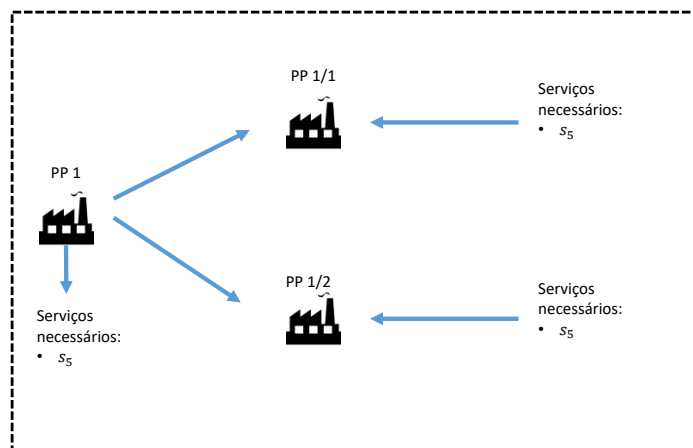


Figura 3.4: Exemplo de divisão de uma fábrica em nós falsos

3.3.2 Pré-processamento da rede

Após criada, a rede tem que ser pré-processada previamente à otimização do modelo, fazendo assim com que existam menos variáveis e menos soluções que se sabem ser não admis-

síveis no contexto do problema, tornando o programa mais rápido e eficaz. Para realizar este pré-processamento foram considerados os conjuntos representados na tabela 3.20.

Tabela 3.20: Conjuntos considerados no pré-processamento da rede

Conjunto	Descrição
R_s	Conjunto de nós iniciais falsos de cada ponto de recolha
R_d	Conjunto de nós falsos onde é realizado o estilhamento de resíduos
R_{ds}	Conjunto de nós falsos onde é realizado o carregamento da estilhadora

Numa primeira fase é necessário garantir que as rotas dos camiões do tipo B previamente determinadas são utilizadas, sendo que para o efeito foi desenvolvido o parâmetro apresentado em 3.21 e a restrição Pré 1.

Tabela 3.21: Novo parâmetro de rotas pré-determinadas

Parâmetro	Descrição
Δ_{ij}	Variável binária com o valor 1 se a rota entre $i \in C$ e $j \in C$ já foi pré-determinada, 0 caso contrário

$$\sum_{s \in S} \sum_{v \in V^b} x_{ijvs} \geq \Delta_{ij} \quad (\text{Pré 1})$$

Contudo existe uma variedade muito grande de opções para processar a rede. As seguintes expressões lógicas foram desenvolvidas ao analisar as soluções obtidas, depois de executado o programa, encontrando erros tanto de fluxos como de determinação de tempos.

- $$\sum_{i \in C} x_{ijvs_3} = 0 \quad \forall j \in C, \forall v \in V^e \quad (\text{Pré 2})$$

- $$\sum_{i \in C} x_{ijvs_8} = 0 \quad \forall j \in C, \forall v \in V^e \quad (\text{Pré 3})$$

- $$\sum_{i \in R_s} x_{ijvs_7} = 0 \quad \forall j \in R_s, \forall v \in V^a \quad (\text{Pré 4})$$

- $$\sum_{i \in R_s} x_{i0vs_{10}} = 0 \quad \forall v \in V^a \quad (\text{Pré 5})$$

- $$x_{ijvs_1} = 0 \quad \forall i \in R_{sd}, \forall j \in R_{sd}, \forall v \in V^a \quad (\text{Pré 6})$$

•

$$t_{ivs_1} \geq Tstart * (\sum_{j \in R_s} x_{jivs_1} * r_{is_1}) \quad \forall i \in C, \forall v \in V^a \quad (\text{Pré 7})$$

•

$$t_{0vs_10} \geq t_{jvs_1} \quad \forall j \in C, \forall v \in V^a \quad (\text{Pré 8})$$

•

$$\sum_{v \in V^a} t_{ivs_1} \geq \sum_{v \in V^b} t_{jvs_4} \quad \forall i \in Rsd, \forall j \in Rsd \quad (\text{Pré 9})$$

A expressão Pré 2 força a 0 as rotas que não são admissíveis entre os diferentes pontos de recolha para as estilhadoras que efetuam o serviço s_3 , isto é, as estilhadoras não se podem mover entre diferentes pontos de recolha sem antes serem transportadas (s_8) para o nó inicial respetivo.

Já a expressão Pré 3 garante que as estilhadoras não executam o serviço s_8 dentro do mesmo ponto de recolha, isto é, no mesmo ponto de recolha não é possível ir de um nó falso para o nó inicial do ponto de recolha.

As expressões Pré 4, Pré 5, e Pré 6 dizem todas respeito a forçar rotas dos camiões do tipo A a 0 quando estas não são admissíveis. A expressão Pré 4 força a que não seja possível ir de um nó inicial de um ponto de recolha diretamente para outro nó inicial diferente. A restrição Pré 5 impede o movimento entre os nós iniciais do e o depósito central. Já a expressão Pré 6 força a 0 as rotas entre os diferentes nós falsos considerados de carga, isto é, o nó falso em cada ponto de recolha que contém o serviço s_1 .

É também necessário restringir tempos para que não existam disparidades nas soluções obtidas em relação às precedências dos tempos. Na expressão lógica Pré 7 é garantido que o serviço s_1 não é efetuado fora dos limites temporais do problema. Com a expressão Pré 8 o retorno para o depósito central por parte dos camiões do tipo A é forçado a ser efetuado apenas no fim da rota, isto é, depois de ser efetuado s_1 .

Por fim ao utilizar a expressão Pré 9 os camiões do tipo A são forçados a efetuar o serviço s_1 apenas depois de todos os serviços de estilhamento s_3 terem sido efetuados, garantido assim que uma vez que uma estilhadora entra num ponto de recolha não sai de lá até terem sido estilhados todos os resíduos.

Com recurso a todo este pré-processamento em conjunto com o pré-processamento já efetuado pelo *software Gurobi* reduz o número de **variáveis binárias** (x_{ijvs}) de **924** para **50** enquanto o número de **variáveis contínuas** é reduzido de **12012** para **138**.

Capítulo 4

Resultados computacionais

Para a implementação do modelo apresentado nesta dissertação foi utilizado o *software Gurobi* em conjunto com o IDE *Spyder* e a linguagem de programação *Python*. No que diz respeito à velocidade de computação, para a instância apresentada, o modelo é otimizado numa média de 7,3 segundos num computador com o sistema operativo *Windows 10*, com 4GB de memória RAM e CPU com frequência de relógio de 2.3 GHz.

4.1 Resolução do caso de estudo

A solução gerada para a instância apresentada anteriormente está representada na tabela 4.1, onde as rotas estão ordenadas de acordo com o seu horizonte temporal, isto é, as rotas entre i e j representadas pela variável x_{ijvs} estão ordenadas do menor para o maior de acordo com o valor da variável t_{jvs} . De notar que na solução apresentada o valor do parâmetro p_{is} toma o valor 1 para todos os serviços, sendo desta forma é possível observar as diferenças nos tempos sem grande esforço. Por ser uma instância de teste e os parâmetros tomarem também valores de teste, o valor da função objetivo não reflete verdadeiramente o custo como é apresentado na descrição da função objetivo.

Tabela 4.1: Solução final ordenada pelos tempos

Veículo	Origem	Destino	Serviço	Execução de s
Estilhadora 2	Depósito	Ponto de recolha 2/0	8	464
Tipo A 2	Depósito	Ponto de recolha 2/0	7	464
Tipo B 1	Depósito	Ponto de recolha 2/1	4	465
Estilhadora 2	Ponto de recolha 2/0	Ponto de recolha 2/1	3	465
Tipo A 1	Depósito	Ponto de recolha 3/0	7	481
Estilhadora 1	Depósito	Ponto de recolha 3/0	8	481
Tipo B 2	Depósito	Ponto de recolha 3/1	4	482
Estilhadora 1	Ponto de recolha 3/0	Ponto de recolha 3/1	3	482
Tipo B 1	Ponto de recolha 2/1	Fábrica 1/2	5	486.556
Tipo A 2	Ponto de recolha 2/0	Ponto de recolha 3/1	1	497
Tipo A 2	Ponto de recolha 3/1	Ponto de recolha 1/0	7	512.556
Estilhadora 1	Ponto de recolha 3/1	Ponto de recolha 1/0	8	512.556
Tipo B 1	Fábrica 1/2	Ponto de recolha 1/1	4	513.556
Estilhadora 1	Ponto de recolha 1/0	Ponto de recolha 1/1	3	513.556
Tipo A 2	Ponto de recolha 1/0	Ponto de recolha 1 /1	1	518.556
Tipo A 1	Ponto de recolha 3/0	Ponto de recolha 2/1	1	523
Tipo B 2	Ponto de recolha 3/1	Fábrica 2/1	5	529.167
Tipo A 1	Ponto de recolha 2/1	Depósito	10	544.556
Estilhadora 2	Ponto de recolha 2/1	Depósito	11	544.556
Tipo A 2	Ponto de recolha 1/1	Depósito	10	544.556
Estilhadora 1	Ponto de recolha 1/1	Depósito	10	544.556
Tipo B 1	Ponto de recolha 1/1	Armazém 2/D1	5	548.556
Tipo B 1	Armazém 2/D1	Armazém 2/S1	9	549.556
Tipo B 2	Fábrica 2/1	Depósito	6	552.167
Tipo B 1	Armazém 2/S1	Fábrica 1/1	5	571.556
Tipo B 1	Fábrica 1/1	Armazém 1/S1	9	626.556
Tipo B 1	Armazém 1/S1	Fábrica 3/1	5	703.556
Tipo B 1	Fábrica 3/ 1	Depósito	6	733.556

O valor da **função objetivo** com a solução apresentada na tabela é de **19978** e, apesar de não representar o valor real dos custos pode ser utilizado para comparar resultados. Após uma análise da solução é possível observar que todos os aspetos de sincronização estão a ser cumpridos.

No que diz respeito à sincronização de movimentos, observa-se que os tempos para os serviços de movimento dos camiões do tipo A estão coincidentes com a estilhadora que estão a transportar, assim como o seu ponto de origem e destino. Na tabela 4.2 está representado um exemplo deste tipo de sincronização, onde se pode observar que os tempos são iguais, assim como a origem e o destino de ambos os veículos enquanto na imagem 4.1 está uma representação gráfica dos valores da tabela onde se pode concluir que os veículos percorrem o mesmo arco ao mesmo tempo cumprindo assim as questões de sincronização de movimento.

Tabela 4.2: Exemplo de sincronização de movimentos

Veículo	Origem	Destino	Serviço	Execução de s
Estilhadora 2	Depósito	Ponto de recolha 2/0	8	464
Tipo A 2	Depósito	Ponto de recolha 2/0	7	464

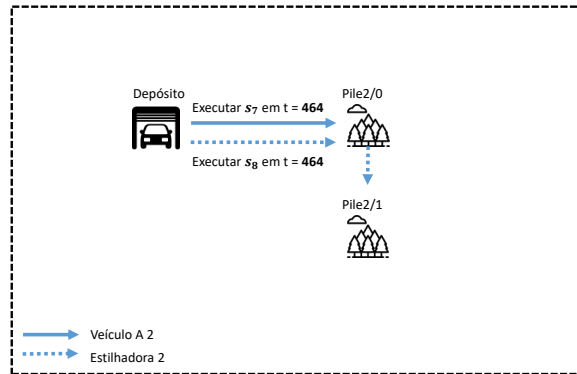


Figura 4.1: Exemplo de sincronização de movimentos

A sincronização de operações com precedência é possível ser observada em todos os tempos apresentados, por exemplo, s_1 só é executado após s_7 pelos camiões do tipo A. Já as precedências nas localizações podem ser observadas com apenas uma unidade de diferença entre os tempos (devido ao parâmetro p_{is} simplificado). Na tabela 4.3 está representado um exemplo de precedência em relação aos veículos, enquanto na tabela 4.4 está representado o único caso de precedência de serviços nas localizações, onde s_3 tem que ser executado antes de s_1 . Na imagem 4.2 podem ser observados dois tipos de sincronização com precedências, isto é, nos veículos e nas localizações. É então possível observar que o veículo A2 só executa o serviço s_1 depois de executar o serviço s_7 e que na localização Pile3/1 é executado o serviço s_3 antes do serviço s_1 .

Tabela 4.3: Exemplo de sincronização de precedência nos veículos

Veículo	Origem	Destino	Serviço	Execução de s
Tipo A 2	Depósito	Ponto de recolha 2/0	7	464
Tipo A 2	Ponto de recolha 2/0	Ponto de recolha 3/1	1	497

Tabela 4.4: Exemplo de sincronização de operações com precedência nas localizações

Veículo	Origem	Destino	Serviço	Execução de s
Estilhadora 1	Ponto de recolha 3/0	Ponto de recolha 3/1	3	482
Tipo A 2	Ponto de recolha 2/0	Ponto de recolha 3/1	1	497

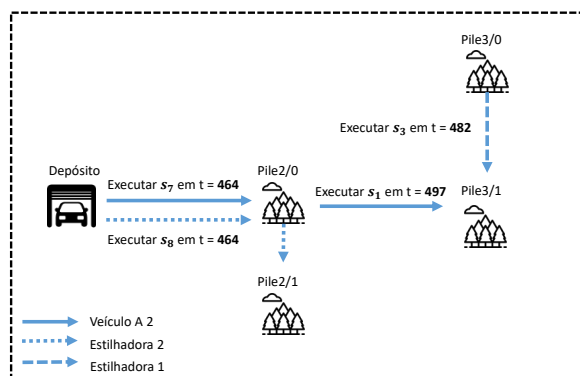


Figura 4.2: Exemplo de sincronização de operações com precedência em veículos e em localizações

Quanto à sincronização de operações simultâneas nas localizações é possível verificar a sua validade de uma forma semelhante à sincronização de movimentos, isto é, os tempos de início dos serviços têm de ser iguais, contudo não é necessário os dois veículos em consideração partilharem o mesmo nó de origem. Por exemplo, o camião do tipo B executa o serviço s_4 ao mesmo tempo que uma estilhadora executa o serviço s_3 na mesma localização mas têm origens diferentes. Na tabela 4.5 está representado um exemplo deste tipo de sincronização onde os serviços s_3 e s_4 são executados na mesma localização ao mesmo tempo sem que os veículos partilhem a mesma origem. Na imagem 4.3 está uma representação gráfica do exemplo da tabela 4.5 onde é possível observar que na Pile2/1 os serviços s_3 e s_4 são executados ao mesmo tempo.

Tabela 4.5: Exemplo de sincronização de operações simultâneas nas localizações

Veículo	Origem	Destino	Serviço	Execução de s
Tipo B 1	Depósito	Ponto de recolha 2/1	4	465
Estilhadora 2	Ponto de recolha 2/0	Ponto de recolha 2/1	3	465

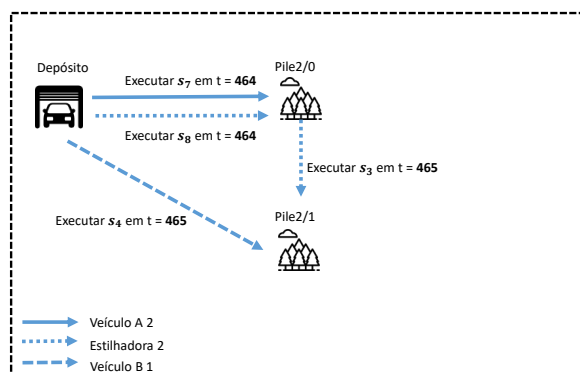


Figura 4.3: Exemplo de sincronização de operações simultâneas em localizações

4.2 Análises complementares

O modelo desenvolvido é considerado uma parte integrante de uma ferramenta de apoio à decisão para o problema descrito anteriormente, assim, é possível realizar algumas análises e comparar resultados obtendo assim uma solução melhor que a obtida anteriormente. Esta análise só é possível porque o modelo apresentado força a que sejam utilizados todos os veículos considerados na instância.

Como exemplo deste tipo de análise foram realizadas três análises com pequenas variações na instância, neste caso nos veículos. Na tabela 4.6 estão representados os três casos que vão ser analisados nesta secção.

Tabela 4.6: Três casos a analisar

Caso	Alteração em relação à instância original
Caso N°1	Menos um veículo do tipo B
Caso N°2	Menos um veículo do tipo A e uma estilhadora
Caso N°3	Menos um veículo do tipo A, tipo B e estilhadora

No primeiro caso foi **eliminado** um dos veículos do **tipo B** e comparado o valor da função objetivo obtida com o que foi obtido anteriormente no secção 4.1. O valor obtido pela função objetivo é de aproximadamente **16411**, pelo resultado obtido é possível afirmar que, com as mesmas localizações da instância apresentada em 3.3, é mais vantajoso utilizar apenas um camião do tipo B. Contudo a utilização de apenas um veículo do tipo B leva a que sejam necessárias aproximadamente mais cem unidades temporais para completar todas os serviços do modelo. Na tabela 4.7 está representada a diferença entre os tempos máximos, isto é o valor máximo da variável t_{ivs} , em cada caso assim como a diferença entre os valores das duas funções objetivo. De notar que nas instâncias testadas todos os veículos têm que ser utilizados, pelo que existem sempre os custos fixos associados a todos os veículos.

Tabela 4.7: Comparação com menos um camião tipo B

	Função objetivo	Tempo máximo
Instância original	19978	733
Instância alterada	16411	838
Diferença de valores	-18%	+14%

Numa segunda fase foi efetuada a mesma análise nas mesmas condições mas desta vez **eliminando** um veículo do **tipo A** e uma **estilhadora** em vez de eliminar um veículo do tipo B. Neste caso a função objetivo toma o valor **14643** e, depois de comparado o valor da função objetivo, é novamente possível afirmar que é mais vantajoso utilizar apenas uma estilhadora e um veículo do tipo A em contraste a utilizar dois veículos de cada tipo. Nesta análise o horizonte temporal da solução não sofre grandes alterações quando comparado com a primeira solução obtida em 4.1, logo

é mais vantajoso utilizar apenas um veículo de cada tipo sem que seja comprometida a rapidez com que os serviços são executados. Na tabela 4.8 é apresentada a diferença das funções objetivo e dos tempos máximos entre os dois casos, que neste caso é desprezável.

Tabela 4.8: Comparação com menos um caminhão A e estilhadora

	Função objetivo	Tempo máximo
Instância original	19978	733
Instância alterada	14646	733
Diferença de valores	-27%	+0%

Por último e com base nas duas situações anteriores, foi realizada uma análise agregando os dois casos anteriores e então verificar se é possível **remover um veículo de cada tipo** (A, B e estilhadora) mantendo a solução válida e com um menor custo. O valor da função objetivo com apenas um veículo do tipo A, um veículo do tipo B e uma estilhadora disponíveis é de **10216**, assim esta opção revela-se mais vantajosa para a resolução do problema e é também um exemplo de como os resultados do problema a um nível operacional podem influenciar os dados a um nível superior, neste caso o número de veículos a utilizar.

Tabela 4.9: Comparação com menos um caminhão de cada tipo

	Função objetivo	Tempo máximo
Instância original	19978	733
Instância alterada	10216	838
Diferença de valores	-51%	+14%

De notar que as soluções destes casos são apresentadas no anexo C da dissertação, sendo assim possível observar que ao alterar a instância todas as sincronizações continuam a ser cumpridas.

Em suma, o modelo apresentado nesta dissertação não só é capaz de resolver o problema de sincronização apresentado inicialmente como também pode ser utilizado como uma ferramenta de análise. Ferramenta esta que ajuda, por exemplo, na decisão da quantidade de veículos a utilizar e pode levar a alterações nos níveis superiores de decisão deste tipo de problemas, isto é, é possível fazer análises do tipo *Trade-off*. Neste caso foi feita uma análise deste género entre o número de veículos e o custo total da utilização dos veículos, ficando claro que o horizonte temporal é limitado pelos camiões do tipo B. Depois desta análise é então possível ao utilizador decidir se é mais vantajoso retirar um veículo de cada tipo, aumentando o horizonte temporal em 14% da solução original mas diminuindo o valor da função objetivo em 49%, ou retirar apenas um veículo do tipo A e uma estilhadora, mantendo assim o horizonte temporal igual à solução original mas diminuindo os custos em apenas 27% da solução original.

Capítulo 5

Interface gráfica de apoio

5.1 Desenvolvimento e apresentação

Para tornar a utilização e testes do modelo desenvolvido mais amigável para o utilizador foi desenvolvida uma ferramenta com recurso ao *software Excel*. Desta forma o utilizador não necessita de ter conhecimentos de programação (nomeadamente na linguagem *Python*) nem de manipulação de dados, para realizar análises simples de forma a encontrar combinações de diferentes parâmetros que melhorem a solução apresentada inicialmente. O ficheiro *Excel* é composto por duas folhas principais, em que uma representa as localizações e a outra representa os veículos. A imagem 5.1 exemplifica os passos que são necessários para a execução programa desenvolvido, assim como a contribuição da interface gráfica desenvolvida.

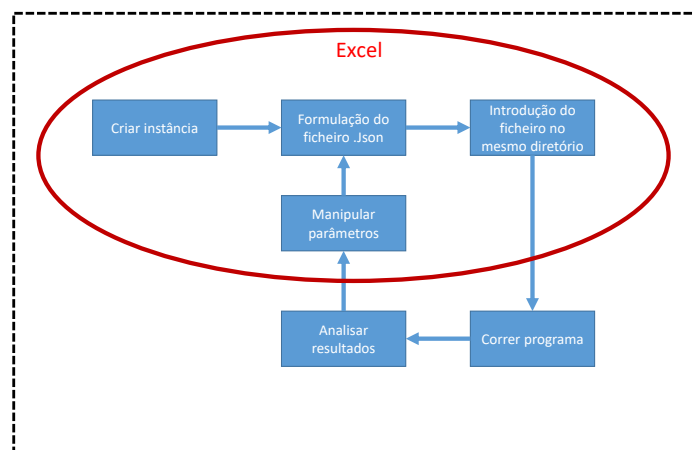


Figura 5.1: Contribuição da interface gráfica.

Com a criação automática do ficheiro *Json* a intervenção por parte do utilizador passa então a ser muito mais simples, representada pela imagem 5.2. Esta automação foi alcançada através da execução de um script *Python* que efetua a leitura dos dados contidos no ficheiro *Excel*, a ser manipulado pelo utilizador, e cria um dicionário que contém os dados com chaves organizadas

de acordo com a estrutura do modelo desenvolvido e implementado nesta dissertação. De referir que a interface referida contém uma variante em *Excel* e *Visual Basic* e outra em *Python* a ser executada pelo mesmo *software* que foi utilizado para a implementação do modelo. A introdução dos dados por parte do utilizador é feita em *Visual Basic* com recurso a *macros*, enquanto a leitura dos dados é feita na linguagem de programação *Python*.

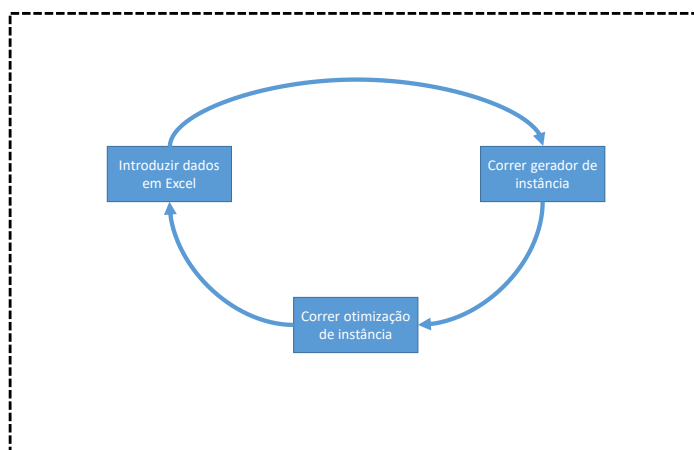


Figura 5.2: Passos a ser executados pelo utilizador com a interface gráfica

A criação das tabelas e das suas dimensões são feitas de forma automática, isto é, os tamanhos das tabelas não são independentes umas das outras o que requer uma alocação dinâmica das mesmas na folha *Excel*. Na figura 5.3 é possível observar que a tabela referente às informações das estilhadoras varia de acordo com o número de pontos de recolha considerados, sendo que esta variação ocorre automaticamente.

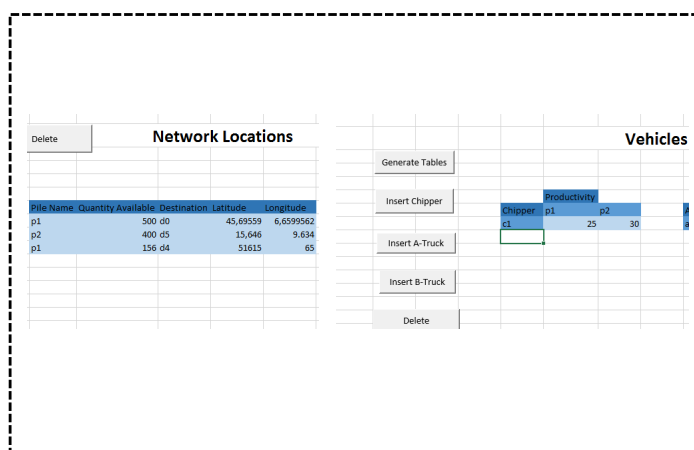


Figura 5.3: Exemplo de alocação dinâmica das tabelas na interface gráfica.

Para além de tudo que já foi referido, a interface desenvolvida também poupa ao utilizador o

cálculo de todas as distâncias entre localizações através do calculo automático das mesmas com base nas coordenadas de latitude e longitude. A figura 5.4 é um pequeno exemplo das distâncias entre alguns pontos considerados na instância, recorrendo às suas respectivas latitude e longitude.



```

44  },
45  "Nodes": {
46    "p1": {
47      "travelDistance": 0
48    },
49    "p2": {
50      "travelDistance": 7989.016665700399
51    },
52    "dep": {
53      "travelDistance": 8288.200823529944
54    },
55    "ter1": {
56      "travelDistance": 4794.747769002549
57    },
58    "p2": {
59      "travelDistance": 7797.001733186055
60    },
61  },
62  "p1": {
63    "p1": {
64      "travelDistance": 7989.016665700399
65    },
66    "p2": {
67      "travelDistance": 0
68    },
69    "dep": {
70      "travelDistance": 11302.852170390377
71    },
72    "ter1": {
73      "travelDistance": 4835.884724118425
74    },
75    "p2": {
76      "travelDistance": 15783.647478217095
77    },
78  },
79  "dep": {
80

```

Figura 5.4: Exemplo de distâncias calculadas automaticamente no ficheiro.

5.2 Manual de utilização

Para uma boa utilização da interface gráfica desenvolvida o utilizador do programa deve ter conhecimento:

- de todas as localizações existentes na instância;
- dos parâmetros das quantidades disponíveis referentes aos pontos de recolha;
- dos parâmetros às capacidades máximas referentes aos armazéns;
- dos parâmetros das capacidades máximas e procura referentes às fábricas;
- da totalidade dos veículos a ser utilizados;
- dos parâmetros referentes aos custos fixos e variáveis veículos do tipo A, assim como as suas capacidades máximas;
- dos parâmetros referentes aos custos fixos e variáveis veículos do tipo B, assim como as suas capacidades máximas;
- dos parâmetros de produtividade em cada ponto de recolha referentes às estilhadoras.

A utilização da interface requer que os **ficheiros** em *Python* (*InstanceCreator.py* e *model.py*) e em *Excel* (*InstanceCreator.msv*) estejam no **mesmo diretório**, assim como a permissão por parte do utilizador à utilização de *macros* no *Excel*.

Para que os dados sejam introduzidos corretamente o utilizador deve começar por criar as tabelas da folha *Locations* do ficheiro *InstanceCreator.msv* com recurso ao botão *Generate Tables*, e após criadas as tabelas o utilizador pode usar os diversos botões de criação de localizações (*Insert Pile*, *Insert Terminal*, *Insert Power Plant*, *Insert Depot*) e então introduzir os dados referentes às mesmas.

As decisões a nível tático pré-determinadas são neste caso representadas na forma de rotas, isto é, cada localização tem pelo menos um destino pré-determinado que um veículo tem que percorrer. Se existir um caso em que exista mais que um destino pré-determinado para a mesma localização inicial então o utilizador deve introduzir a mesma localização mais uma vez com o mesmo nome e restantes dados e apenas mudar a coluna *Destination*.

Após introduzidos todos os dados referentes às localizações da instância o utilizador deve ir para a folha *Vehicles* e gerar os cabeçalhos das tabelas referentes aos veículos. Só então é possível introduzir todos os dados referentes aos veículos com recurso aos diversos botões para o efeito (*Insert A-Truck*, *Insert B-Truck*, *Insert Chipper*).

De referir que ambas as folhas do ficheiro *Excel* contêm um botão para apagar todo o conteúdo da página e assim recomençar a construir a instância.

Após introduzidos todos os dados referentes à instância o utilizador deve correr o ficheiro *InstanceCreator.py*, após a fim do programa o resultado é um ficheiro *Json* que contém todos os dados da instância. De notar que dados como o horizonte temporal e os serviços considerados não são representados na interface pois não são normalmente alterados com bastante frequência, este tipo de alterações pode ser efetuada no ficheiro *InstanceCreator.py*.

É então possível correr o programa que contém o modelo desenvolvido e analisar os resultados referentes à instância criada através da utilização do ficheiro *model.py*.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusões e contribuições do trabalho desenvolvido

O trabalho desenvolvido nesta dissertação teve como origem um problema real de uma empresa baseada na Finlândia. O objetivo principal da dissertação é a otimização de rotas de veículos num contexto de recolha e transporte de biomassa a um nível operacional, com alguns pontos pré-determinados a nível tático. Neste enquadramento são considerados diferentes tipos de sincronização que também requerem otimização em termos de custos e janelas temporais. O problema em estudo é classificado como um problema de planeamento de rotas de veículos com múltiplas restrições de sincronização (VRPMS - *Vehicle Routing Problem with Multiple Synchronization Constraints*). Importante referir que a revisão bibliográfica sobre o tema em questão revelou que não existe muita variedade em termos de literatura neste tipo de problemas onde existem múltiplos tipos de sincronização, é então necessária uma forma diferente de encarar o problema.

Foi inicialmente proposta uma abordagem orientada aos serviços, baseada num modelo matemático apresentado por Mankowska num problema de planeamento de rotas e intervalos temporais no contexto da prestação de cuidados de saúde em casa (HHCSP - *Home Health Care Routing and Scheduling Problem*). Uma abordagem orientada aos serviços permitiu uma maior abstração do modelo, e assim a sua possível implementação noutros contextos. A adaptabilidade da abordagem só é possível pois os serviços podem ter o significado que o utilizador quiser, nesta dissertação o significado dos serviços varia desde um veículo em transporte até ao carregamento de resíduos florestais. Para além de uma maior abstração do problema é também possível garantir os tipos de sincronização necessários neste contexto, pois ao considerar serviços é possível manipular facilmente os seus intervalos temporais ao utilizar restrições referentes a pares de serviços, e assim obter as sincronizações necessárias.

Após desenvolvido o modelo com a abordagem referida é então necessário realizar testes numa instância que possa mais tarde voltar a ser testada no mesmo contexto com uma abordagem diferente para que seja possível comparar os resultados de eficiência do modelo desenvolvido nesta dissertação. A instância proposta tem uma dimensão relativamente pequena quando comparada a

um problema de dimensão real, pelo que os testes realizados constituem uma prova de conceito da aplicação do modelo desenvolvido.

Para implementar o modelo proposto e alcançar todos os tipos de sincronização, assim como outros pontos que advêm de um nível tático do problema, foi necessária a criação da instância e da sua respetiva rede de localizações. Após criada, a rede requer um pré-processamento complexo que envolve a divisão dos diferentes tipos de localizações em sub-localizações com diferentes categorias e funcionalidades. As questões táticas referidas também têm de ser pré-processadas ao forçar certas rotas no programa desenvolvido, sob a forma de restrições e equações lógicas.

Como resultado do modelo todos os pontos de sincronização foram cumpridos e todos os serviços considerados cumpridos, assim o programa desenvolvido pode ser integrado numa ferramenta de apoio à decisão e à otimização do planeamento de rotas de veículos, reforçando a ideia de adaptabilidade de um modelo orientado aos serviços. Para além de cumpridos todos os requisitos e otimizadas as rotas dos veículos na instância criada, foi também realizada uma série de análises com a simultânea manipulação das diferentes quantidades dos veículos considerados. Estas análises permitiram uma combinação de resultados que permite ao utilizador escolher diferentes soluções para o mesmo problema, cada uma com a sua vantagem em termos de tempo ou de custo.

A ferramenta de apoio desenvolvida tem como suplemento uma interface gráfica em *Excel* que permite a criação de uma instância com a estrutura preparada para o código em *Python* desenvolvido nesta abordagem. A interface gráfica é composta por botões que induzem o utilizador a introduzir os dados referentes à instância que vai ser criada. A interface gráfica permite um grande distanciamento do utilizador das questões de programação e assim tornar a ferramenta mais amigável para o utilizador.

Em suma, pode-se concluir que todos os objetivos iniciais foram cumpridos e que a ferramenta e o modelo desenvolvido cumprem todas as questões levantadas e a abordagem orientada aos serviços é, de facto, uma mais valia num contexto de transporte e recolha de biomassa assim como numa variedade de diferentes contextos. Futuramente a integração do que foi desenvolvido numa plataforma com maior dimensão que permita uma maior abstração e otimização das rotas dos veículos pode constituir uma mais valia para a área.

6.2 Trabalho Futuro

O modelo desenvolvido nesta dissertação, apesar de cumprir todos os requisitos propostos inicialmente e ter o resultado esperado, não está completo e necessita de alguns ajustes no futuro. O modelo desenvolvido é adequado para a resolução deste tipo de problemas em instâncias relativamente pequenas e, assim não pode ser utilizado diretamente numa empresa do mesmo contexto. É então necessária a realização de testes em instâncias de maior dimensão assim como os consequentes ajustes ao modelo que irão ser necessários.

Para além da questão da dimensão da instância é possível melhorar a dissertação através de um maior pré-processamento do modelo, tornando assim o programa mais rápido e eficiente, sendo aspeto muito importante para o modelo ser testado numa dimensão maior. Para além do

pré-processamento adicional também é possível melhorar os tempos computacionais através da exploração e implementação de diferentes métodos de resolução no problema.

No que diz respeito à ferramenta gráfica desenvolvida em *Excel*, apesar de altamente funcional, é possível melhorar amplamente o seu aspeto gráfico, e assim tornar a experiência de utilização ainda mais agradável. Para além de uma melhoria no aspeto gráfico, a implementação do cálculo das distâncias diretamente no ficheiro *Excel* também é um aspeto que melhora a experiência de utilização da ferramenta gráfica. Na mesma lógica também é possível implementar um mapa, recorrendo a API's da *Google*, e assim permitir ao utilizador escolher as suas localizações através de um mapa em vez da introdução das coordenadas latitude e longitude de cada localização.

Anexo A

Código *Python* da implementação do modelo

Neste anexo é apresentado todo o código devidamente comentado que foi implementado na dissertação.

A.1 Código do modelo

```
# -*- coding: utf-8 -*-  
"""
```

2017-06-25

BIOMASS SUPPLY ROUTING AND SCHEDULING

Main Assumptions

```
"""
```

```
#####  
####LIBRABRIES / PACKAGES TO BE IMPORTED####  
#####  
import json
```

```

import numpy as np
from gurobipy import *

#-----#

#####
#####SCRIPT CONFIG & DATA I/O#####
#####

#Specifying Instance Name
instance = 'instance_2'
#Specifying Model Name for Gurobi
model_name = instance
#Path for JSON file#
inputdata_path = 'json/' + instance + '.json'
#Path for Output LP File
lp_path = 'lp/' + instance + '.lp'
#Path for Output MPS File
mps_path = 'mps/' + instance + '.mps'
#Path for Output SOL File
sol_path = 'sol/' + instance + '.sol'
#Path for a simplified txt with solution
txtsol_path = 'txtsol/' + instance + '.txt'
#Path for Output ILP File (is computed if model is infeasible and gurobi_genIIS
ilp_path = 'ilp/' + instance + '.ilp'

####GUROBI PARAMETERS/SETTINGS###
#Path for Gurobi Log File
gurobi_logpath = 'gurobi_log/' + instance + '.log'
#Time Limit in minutes (default 1e+100 seconds)
gurobi_TimeLimit = 1
#Maximum MIP Relative Gap for Gurobi Solver (default 1e-4)
gurobi_MIPGap = 1e-4
#Maximum MIP Absolute Gap for Gurobi Solver (default 1e-10)
gurobi_MIPGapAbs = 1e-10
#NumericFocus Parameter for numerical accuracy (default 0 = auto)
gurobi_NumericFocus = 0
#Percentage of time devoted to heuristics (default 0.05)

```

```

gurobi_Heuristics = 0.05
#Perform optimization , or just import instance
gurobi_optimize = True
#Generate Irreducible Inconsistent Subsystem (IIS) if model is infeasible or
gurobi_genIIS = True
#Print statistics about the quality of the computed solution
gurobi_printQuality = True
#Reset the model to an unsolved state after optimizing
gurobi_resetOnEnd = True
#Settings for defining SOS constraints
gurobi_SOS = False

#Opening JSON File
with open(inputdata_path) as data_file:
    data = json.load(data_file)

#Special character that delimits dummy nodes in string
#(CANNOT BE USED IN ANY OTHER SITUATIONS)
specialChar = '/'

#-----#

##Auxiliary function
#Get real-life node from a dummy node
def realNode(dummyNode):
    if dummyNode.rpartition(specialChar)[0] == '':
        return dummyNode
    else:
        return dummyNode.rpartition(specialChar)[0]

#####
#####SETS#####
#####

###MAIN SETS###

#Set of chippers

```

```

K = list(data['Chipper'].keys())

#Set of A-trucks (haulers) that transport chippers
H = list(data['A-Truck'].keys())

#Set of B-trucks (haulers) that transport wood chips
L = list(data['B-Truck'].keys())

#Set of raw material piles at the roadside
P = list(data['Pile'].keys())

#Set of dummy nodes in Piles

R = [p + '/' + str(i)
      for p in P
      for i in range(len(data['Pile'][p]['dummyNodes'])+1)]
#Set of start dummy nodes
Rs = [r
      for r in R
      if r == ('pile1/0') or r == ('pile2/0') or r == ('pile3/0')]
#set of chipping dummy Nodes
Rd = R
for r in Rs:
    Rd.remove(r)
#Set of depots, it is assumed to only exist one
O = list(data['Depot'].keys())

#Set of power plants

M = list(data['PowerPlant'].keys())
#Set of power plants' dummy
Mstar_m = dict((m, []) for m in M)

Pstar_p = dict((p, [(str(p) + specialChar + str(i+1)) for i in range(len(data['Pile'][p]['dummyNodes'])+1)]) for p in P)
for p in P:
    Pstar_p[p].append(str(p) + specialChar + '0')
del p

#Set of terminals

```

```

S = list(data['Terminal'].keys())

#Set of terminal dummy demand
#Set of terminals' dummy (supply and demand) nodes

Sdstar_s = dict((s, []) for s in S)
Ssstar_s = dict((s, []) for s in S)
#These dummy nodes will be determined simultaneously
#and (predetermined) routes will already be imported
routes = []

for s in S:
    for i in data['Terminal'][s]['dummyNodes']:
        aux1 = str(s) + specialChar + 's' + str(len(Ssstar_s[s]) + 1)
        aux2 = str(i[0]) + specialChar + str(len(Mstar_m[i[0]]) + 1)
        routes.append([aux1, aux2])
        Ssstar_s[s].append(aux1)
        Mstar_m[i[0]].append(aux2)
        del aux1, aux2

for p in P:
    for i in range(len(Pstar_p[p]) - 1):
        aux1 = data['Pile'][p]['dummyNodes'][i][0]
        aux3 = str(p) + specialChar + str(i + 1)
        if aux1 in S:
            aux2 = str(aux1) + specialChar + 'd' + str(len(Sdstar_s[aux1]) + 1)
            routes.append([aux3, aux2])
            Sdstar_s[aux1].append(aux2)
        elif aux1 in M:
            aux2 = str(aux1) + specialChar + str(len(Mstar_m[aux1]) + 1)
            routes.append([aux3, aux2])
            Mstar_m[aux1].append(aux2)
        del aux1, aux3
    del p, i

#Set of Services
X = list(data['Service'].keys())

```

```

#Set of services that vehicle v may start its work
G = dict((v, data['A-Truck'][v]["sinit"]) for v in H)
for v in K:
    G[v] = data['Chipper'][v]["sinit"]
for v in L:
    G[v] = data['B-Truck'][v]["sinit"]

#Set of ordered pairs of services that require v to have immediately performed
#service s1 before s2
A = dict((v, data['A-Truck'][v]["sdouble"]) for v in H)
for v in K:
    A[v] = data['Chipper'][v]["sdouble"]
for v in L:
    A[v] = data['B-Truck'][v]["sdouble"]

#Set of ordered pairs of services that require costumer i
#to receive s1 before s2
B = dict((p,0) for p in Rd)
#Set of end Pile dummy Nodes
Rsd = list()
for p in P:
    for i in range(len(data['Pile'][p]['dummyNodes']) + 1):
        if (i == len(data['Pile'][p]['dummyNodes'])):
            B[p+'/' + str(i)] = data['Pile'][p]["sprec"]
            Rsd.append(p + '/' + str(i))

#Set of ordered pairs of services that require costumer i
#to receive s1 at the same time as s2
C = dict((p,0) for p in R)
for p in Rs:
    C[p] = data['Pile'][realNode(p)]["ssimul"][1]
for p in Rd:
    C[p] = data['Pile'][realNode(p)]["ssimul"][0]
    C['dep'] = data['Depot']['dep']['ssimul']
#Set de todos os veiculos

D = list(set(K) | set(H) | set(L))

#Set de todas as localiza es

```



```

E = list(set(P)| set(S) | set(M) | set(O))

#Complete set of Nodes for dummy Nodes in each location
Sdstar = [Sdstar_s[s][i] for s in S for i in range(len(Sdstar_s[s]))]
Ssstar = [Ssstar_s[s][i] for s in S for i in range(len(Ssstar_s[s]))]
Mstar = [Mstar_m[m][i] for m in M for i in range(len(Mstar_m[m]))]
Pstar = [Pstar_p[p][i] for p in P for i in range(len(Pstar_p[p]))]
#####
####PARAMETROS#####
#####

#a_vs -> 1 se o ve culo v pode executar o servi o 0 ou 0 se n o .
a_vs=dict(((v,s),0)for v in D for s in X)
for v in D:
    if v in K:
        for s in data['Chipper'][v]['allservices']:
            a_vs[v,s] = 1
    if v in H:
        for s in data['A-Truck'][v]['allservices']:
            a_vs[v,s] = 1
    if v in L:
        for s in data['B-Truck'][v]['allservices']:
            a_vs[v,s] = 1

#r_is -> 1 se a localiza o i necessita do servi o s
Nodes = list(set(R)| set(Ssstar) | set(Sdstar) | set(O) | set(Mstar))
r_is= dict(((i,s),0)for i in Nodes for s in X)
for i in E:
    for f in R:
        if realNode(f) in P:
            for s in X:
                if s in data['Pile'][realNode(f)]['allservices']:
                    r_is[f,s] = 1
                else:
                    r_is[f,s] = 0
    if i in S:
        for s in data['Terminal'][i]['allservices']:
            r_is[i,s] = 1
    if i in M:

```

```

        for s in data['PowerPlant'][i]['allservices']:
            r_is[i,s] = 1
    if i in O:
        for s in data['Depot'][i]['allservices']:
            r_is[i,s] = 1

#r_is dividido para os dummy Nodes ( pilhas / terminais)
for p in P:
    for i in range(len(data['Pile'][p]['dummyNodes'])+1):
        for s in X:
            if s in data['Pile'][p]['allservices']:
                r_is[p + '/' + str(i),s] = 1
                if (i == 0 and s == 's4'):
                    r_is[p + '/' + str(i),s] = 0
                if (i == 0 and s=='s3'):
                    r_is[p + '/' + str(i),s] = 0
                if (i == 0 and s=='s1'):
                    r_is[p + '/' + str(i),s] = 0
                if (i >0 and (s == 's2' or s == 's7' or s == 's8')):
                    r_is[p + '/' + str(i),s] = 0
                if ((i < len(data['Pile'][p]['dummyNodes'])) and s == 's1'):
                    r_is[p + '/' + str(i),s] = 0
            else:
                r_is[p + '/' + str(i),s] = 0

for p in Sdstar:
    r_is[p,'s5'] = 1
for p in Ssstar:
    r_is[p,'s9'] = 1

for p in Mstar:
    r_is[p,'s5'] = 1

##Time distance between location i and location j
def d_ij(i, j, time = True):
    if time:
        return data['Nodes'][realNode(i)][realNode(j)]['travelTime']
    else:
        return data['Nodes'][realNode(i)][realNode(j)]['travelDistance']

```

```

#c_v -> custo de ve c u l o por unidade temporal e c_vF -> custo fixo por ve i
c_v = dict(((v),0)for v in D)
c_vF = dict(((v),0)for v in D)
for v in D:
    if v in K:
        c_v[v] = data['Chipper'][v]['transportCost']
        c_vF[v] = data ['Chipper'][v]['fixedCost']
    if v in H:
        c_v[v] = data['A-Truck'][v]['transportCost']
        c_vF[v] = data ['A-Truck'][v]['fixedCost']
    if v in L:
        c_v[v] = data['B-Truck'][v]['transportCost']
        c_vF[v] = data ['B-Truck'][v]['fixedCost']

#p_is -> dura o do servi o s na localiza o i assumido 1 para todos os
p_is= dict(((i,s),1)for i in E for s in X)
#p_iks -> dura o do servi o s na localiza o i pela chipper k
p_ik=dict(((i,k),0)for i in R for k in K)
SS = list(set(Ssstar)| set(Sdstar))
for p in P:
    for i in range(len(data['Pile'][p]['dummyNodes'])+1):
        for s in X:
            p_is[p+'/' +str(i),s] = 1

for p in SS:
    for s in X:
        if p in Sdstar:
            p_is[p,s] = 1
        elif p in Ssstar:
            p_is[p,s] = 1
for p in M:
    for i in range(len(Mstar)):
        for s in X:
            p_is[p+'/' +str(i+1),s] = 1

for p in P:
    for i in range(len(data['Pile'][p]['dummyNodes'])):
        for k in K:
            p_ik[p+'/' +str(i+1),k] = data['Pile'][p]['dummyNodes'][i][1] / d

```

```

##-----#
#
#####
##### Cria o da rede#####
#####
network = list()

Nodes = list(set(R)| set(Ssstar) | set(Sdstar)| set(O) | set(Mstar)| set(Rs))
VAK = list(set(K)| set(H))
#arcos entre depot e todos os n s
for i in Nodes:
    for j in Nodes:
        for v in D:
            for s in X:
                if j != i :
                    network.append({'origin ': i,
                                    'destination ': j,
                                    'vehicle ': v,
                                    'service ': s})

#Timeline start / end time
Tstart = data['Timeline']['startTime']
Tend = data['Timeline']['endTime']
#
#
##-----#

#####
#####INSTANTIATION of GUROBI MODEL#####
#####

#Creating model object
env = None
env = Env(gurobi_logpath)
model = None

```

```

model = Model(model_name, env)
model.message("*****MODEL " + model_name + "*****")
#Setting model parameters
model.setParam('TimeLimit', gurobi_TimeLimit * 60)
model.setParam('NumericFocus', gurobi_NumericFocus)
model.setParam('MIPGap', gurobi_MIPGap)
model.setParam('MIPGapAbs', gurobi_MIPGapAbs)
model.setParam('Heuristics', gurobi_Heuristics)

#-----#

#
#####
#####DECISION VARIABLES#####
#####

model.message("Variáveis de decisão")

#Binary variables for routing/scheduling

aux_vsi=dict(((i,j,v,s),a_vs[v,s]*r_is[j,s]) for v in D for s in X for i in I)
model.message("Creating vars x[i,j,v,s]... ")
x_ijvs = model.addVars([(net['origin'],
                           net['destination'],
                           net['vehicle'],
                           net['service'])
                        for net in network],
                        vtype = GRB.BINARY,
                        lb = 0,
                        ub = aux_vsi,
                        name = 'x')

##Variável de decisão tempo de início do serviço do pelo veículo s na I
model.message("Creating vars t[i,v,s]... ")
t_ivs = dict()
for i in Nodes:
    for v in D:
        for s in X:

```



```

        for v in K
    )
        , name = "CPRE2")
#for ar rotas entre pilhas a 0 para s8
#Pr processamento da rede de rotas para chippers
model.message(" Restri o Pr processamento 3")
model.addConstrs((quicksum(x_ijvs[i,j,v,"s8"]
        for i in Rd
        if i !=j
        if realNode(i)==realNode(j))
        ==0
        for j in Rs
        for v in K),name = "CPRE3")
#for ar rotas entre pilhas start a 0 para s7
#Pr processamento da rede de rotas para A trucks
model.message(" Restri o Pr processamento 4")
model.addConstrs((quicksum(x_ijvs[i,j,v,"s7"]
        for i in Rs
        if i !=j
        if realNode(i)!=realNode(j))
        ==0
        for j in Rs
        for v in H),name = "CPRE4")
#for ar rotas entre pilhas/0 e dep a 0 para s10
#Pr processamento da rede de rotas para A trucks
model.message(" Restri o Pr processamento 5")
model.addConstrs((quicksum(x_ijvs[i,'dep',v,"s10"]
        for i in Rs
        if i != j)
        == 0
        for v in H), name = "CPRE5")
#for ar rotas entre n s de carga (pilhas) a 0 para s1
#Pr processamento da rede de rotas para A trucks
model.message(" Restri o Pr processamento 5")
model.addConstrs((x_ijvs[i,j,v,"s1"]
        == 0
        for i in Rsd
        for j in Rsd
        if i != j
        for v in H

```

```

        ), name = "CPRE6")
#for ar que t,s1 seja maior que Tstart
#Pr processamento de tempos de A trucks
model.message(" Restri o Pr processamento 6")
model.addConstrs((t_ivs[i,v,"s1"]>= Tstart*( quicksum(x_ijvs[j,i,v,"s1"]*r_is[i,
        for j in Rs
        if i !=j
        ))
        for i in Rsd
        for v in H
        )
        ,name = "CPRE7")
#for ar que o retorno para o depot seja efetuado no fim da rota
#Pr Processamento de tempos de Atrucks
model.message(" Restri o Pr processamento 7")
model.addConstrs((t_ivs['dep',v,s] >= t_ivs[j,v,s2]
        for v in H
        for s in X
        for s2 in X
        for j in Rsd
        if s == "s10"
        if s2 == "s1"), name ="CPRE8")

model.message(" Restri es ")
#Todas os ve culos que saem do depot t m que voltar
#Restri ao 1
model.message(" Restri ao 1")
del j, i ,v, s
model.addConstrs(( quicksum(x_ijvs['dep',i,v,G[v]]
        for i in Nodes
        if i != 'dep')
        ==
        quicksum(x_ijvs[i,'dep',v,s]
        for i in Nodes
        for s in X
        if i != 'dep')
        for v in D)
        , name = 'CR01')
```



```

#Nenhum material perdido nos nós (inflow-outflow) ( tudo que entra num nó é o que sai)
#Restrição 2
model.message("Restrição 2")
model.addConstrs((quicksum(x_ijvs[i,j,v,s]
                        for i in Nodes
                        for s in X
                        if i != j)
                ==
                quicksum(x_ijvs[j,i,v,s]
                        for i in Nodes
                        for s in X
                        if i != j)
                for v in D
                for j in Nodes)
                , name = 'CR02')

#Garantir a atribuição de apenas 1 veículo que pode executar o serviço
#Formar a 1 uma das rotas possíveis para efetuar o serviço
#Restrição 3
model.message("Restrição 3")
model.addConstrs((quicksum(a_vs[v,s]*x_ijvs[j,i,v,s]
                        for v in D
                        for j in Nodes
                        if i != j
                        )
                ==
                r_is[i,s]
                for s in X
                for i in Nodes
                if s != "s6"
                if i != "dep"
                )
                , name = 'CR03')

#As rotas que contêm os serviços de retorno ao depot (s6,s10,s11) são formadas por B-truck
#B-truck
model.addConstrs((quicksum(a_vs[v,"s6"]*x_ijvs[j,"dep",v,"s6"]
                        for j in Nodes
                        if j != "dep")
                ==
                r_is["dep","s6"]
                for v in L)

```

```

        , name = 'CR03.1')

#A-truck
model.addConstrs((quicksum(a_vs[v,"s10"]*x_ijvs[j,"dep",v,"s10"]
                        for j in Nodes
                        if j != "dep")
                >=
                r_is["dep","s10"]
                for v in H)
        , name = 'CR03.2')

#chipper
model.addConstrs((quicksum(a_vs[v,"s11"]*x_ijvs[j,"dep",v,"s11"]
                        for j in Nodes
                        if j != "dep")
                ==
                r_is["dep","s11"]
                for v in K)
        , name = 'CR03.3')

#Garantir que o servi o inicial de cada servi o o primeiro a ser executado
#Restri o 4
model.message("Restri o 4")
model.addConstrs((Tstart
                + d_ij('dep',j)
                <= t_ivs[j,v,G[v]]
                + Tend*(1-x_ijvs['dep',j,v,G[v]])
                for j in Nodes
                for v in D
                if j != 'dep') , name = "CR04" )

#Sets auxiliares
#dep + pile
DpP = list(set(R)|set(Rs)|set(O))
#pile_dummy + pile start
DupS = list (set(Rs)|set(R))
RsO = list(set(Rs)|set(O))
#Preced ncia de ve culos – o Ve culo tem de executar s1 antes de s2
#Restri o 5
model.message("Restri o 5")
#Atruck
for v in H:
    for z in range(len(A[v])):

```



```

, name = "CR05 - B truck.3" + '[' +
+ str(i) + '/' +
+ str(j) + '/' +
+ str(v) + '/' + ']'

if aux2[0] == 'ter2':
    for i in Nodes:
        for j in Nodes:
            if i != j:
                model.addConstr(( t_ivs[i,v,aux2[1]]
                                + p_is[i,aux2[1]]
                                + d_ij(i,j)
                                <=
                                t_ivs[j,v,aux2[2]]
                                + Tend*(2-x_ijvs[i,j,v,aux2[2]]
                                - quicksum(x_ijvs[z,i,v,aux2[2]]
                                for z in Nodes
                                if z != i)

                                )
                                , name = "CR05 - B truck.4" + '[' +
                                + str(i) + '/' +
                                + str(j) + '/' +
                                + str(v) + '/' + ']'

if aux2[0] == 'pp':
    for i in Nodes:
        for j in Nodes:
            if i != j:
                model.addConstr(( t_ivs[i,v,aux2[1]]
                                + p_is[i,aux2[1]]
                                + d_ij(i,j)
                                <=
                                t_ivs[j,v,aux2[2]]
                                + Tend*(2-x_ijvs[i,j,v,aux2[2]]
                                - quicksum(x_ijvs[z,i,v,aux2[2]]
                                for z in Nodes
                                if z != i)

                                )
                                , name = "CR05 - B truck.5" + '[' +
                                + str(i) + '/' +

```

```

+ str(j) + '/'
+ str(v) + '/' + ']'')

#chippers

for v in K:
    for z in range(len(A[v])):
        aux = A[v]
        aux2 = aux[z]
        if aux2[0] == 'piledum':
            for i in Rs:
                for j in Rd:
                    if (realNode(i) == realNode(j)):
                        model.addConstr((t_ivs[i,v,aux2[1]]
                                         + p_is[i,aux2[1]]
                                         + d_ij(i,j)
                                         <=
                                         t_ivs[j,v,aux2[2]]
                                         + Tend*(2-x_ijvs[i,j,v,aux2[1]]
                                         - quicksum(x_ijvs[z,i]
                                                    for z in Rs
                                                    if z != i)
                                         )
                                         , name = "CR05 - Chipper.1"+ '['
                                         + str(i) + '/'
                                         + str(j) + '/'
                                         + str(v) + '/' + ']'')

        if aux2[0] == 'dpp':
            for i in Rd:
                for j in list(set(Rs)|set(O)):
                    if i != j:
                        if realNode(i) != realNode(j):
                            model.addConstr((t_ivs[i,v,aux2[1]]
                                                 + p_ik[i,v]
                                                 + d_ij(i,j)
                                                 <=
                                                 t_ivs[j,v,aux2[2]]
                                                 + Tend*(2-x_ijvs[i,j,v,aux2[1]]
                                                 - quicksum(x_ijvs[z,i]
                                                            for z in Rs
                                                            if z != i)
                                                 )

```



```

for z in Nodes
    if z != i)

        )

    )
    , name = "CR05 – Chipper.4"+'['
    +str(i) + '/'
    + str(j) + '/'
    + str(v) + '/' + ']'')

#restrição 3.7 -> q_sls2 = 1 -> precedência
#Set auxiliar
F = list(set(M) | set(S) | set(O))
#Precedência de serviços nas localizações (tem de ser executado sl antes)
#Restrição 6
model.message("Restrição 6")
for i in Rsd:
    for v in K:
        for z in H:
            if v != z:
                aux = B[i][0]
                if aux[0] != aux[1]:
                    model.addConstr((t_ivs[i,z,"s1"] - t_ivs[i,v,"s3"]
                                     >= p_ik[i,v] - Tend*(2 -
                                     quicksum(x_ijvs[j,i],
                                               for j in Nodes
                                               if i != j)
                                     + quicksum(x_ijvs[j,i],
                                               for j in Nodes
                                               if i != j)
                                     )

                    )

        )

    ,name = "CR06" + '/' +str(i) + '/' +str(v) + '/' +str(z))

#Simultaneidade de serviços nas localizações (sl tem de ser executado ao mesmo tempo)
#Restrição 7
model.message("Restrição 7")
#Processamento dos tempos, o tempo de início chipping (s3) tem que ser

```

```

model.addConstrs((quicksum(t_ivs[i,k,"s3"]
                           for k in K)
                 ==
                 quicksum(t_ivs[i,l,"s4"]
                           for l in L)
                 for i in Rd), name = "CPRE9")

del i
#Pr processamento dos tempos, a chipper s carregada (s1) depois de ter sido
model.addConstrs((quicksum(t_ivs[i,h,"s1"]
                           for h in H)
                 >= quicksum(t_ivs[j,k,"s4"]
                           for k in L)
                 for i in Rsd
                 for j in Rsd
                 if realNode(i)==realNode(j)), name = "CPRE10")

#Pr processamentos das rotas de A truck, nao pode fazer s7 de /1 para /0
del j, v
model.addConstrs((x_ijvs[i,j,v,"s7"]
                 == 0
                 for i in Rsd
                 for j in Rs
                 for v in H
                 if realNode(i)==realNode(j)
                 ), name = "CPRE11")

#Garantir s7 e s8 ocorrem ao mesmo tempo (quase) e que partem do mesmo ponto
for i in Rs:
    for v in H:
        for z in K:
            if z != v:
                aux2 = C[i]
                if aux2[0] != aux2[1]:
                    model.addConstr((t_ivs[i,z,'s8'] - t_ivs[i,v,'s7']
                                     >= -Tend*(2 -
                                     (quicksum(x_ijvs[j,i,v,'s7']+x_ijvs[j,i,v,'s8']
                                                for j in Nodes
                                                if i != j)
                                     )

```

```

    )
    )
    ,name = "CR07.3" + '[' + str(i) + ',' + str(i) + ',' + str
']')

```

```

#Garantir s10 e s11 ocorrem ao mesmo tempo (quase) e que partem do mesmo ponto
for i in O:
    for v in H:
        for z in K:
            if z != v:
                aux2 = C[i]
                if aux2[0] != aux2[1]:
                    model.addConstr((t_ivs[i,z,'s11'] - t_ivs[i,v,'s10'])
                                     == -Tend*(2 -
                                     quicksum(x_ijvs[j,i,v,'s10']
                                              +x_ijvs[j,i,z,'s11']
                                              for j in Nodes
                                              if i != j)
                                     )
                    ,name = "CR07.4" + '[' +str(i) + ',' + str(i) + ',' + str(i) + ']'

```

```
#Domínio das variáveis de decisão t_ivs
#Restrição 8
model.message("Restrição 8")
model.addConstrs((a_vs[v,s]*r_is[i,s]*Tstart
                  <= t_ivs[i,v,s]<= a_vs[v,s]*r_is[i,s]*Tend
                  for v in D
                  for i in Nodes
                  for s in X)
                  , name = "CR08" )
```

```
# Restrição que garante liga o entre t_ivs e x_jivs
# Restrição 9
model.message("Restrição 9")
model.addConstrs((t_ivs[i,v,s]
```

```

<= Tend*quicksum(x_ijvs[j,i,v,s]
                  for j in Nodes
                  if i != j )

for i in Nodes
for v in D
for s in X
)
, name = "CR09")

#Garantir que o tempo de descarga(s4) igual ao tempo de chipping (s3) ( dura
#Restri o 10
model.message(" Restri o 10")
model.addConstrs((t_ivs[i,b,"s4"] + quicksum(x_ijvs[z,i,b,"s4"]*p_ik[i,k]
                                              for z in Nodes
                                              if i !=z) + d_ij(i,j)

<=
t_ivs[j,b,"s5"] +
Tend*(2- x_ijvs[i,j,b,"s5"] - quicksum(x_ijvs[z,i,k,
                                          for z in Nodes
                                          if i != z))

for b in L
for i in Rd
for j in Nodes
for k in K
if i != j), name = "CR10")

#####
#####OBJECTIVE FUNCTION#####
#####
#
model.update()

objective = GRB.MINIMIZE

totalTime = LinExpr(quicksum(c_v[v]*t_ivs['dep',v,s]
                           for v in D for s in X) +
                    quicksum(c_vF[v]*x_ijvs[i,'dep',v,s]
                           for i in Nodes for v in D for s in X if i != 'dep')
                    )

model.setObjective(totalTime , objective)

```

```

model.update()
model.message("Printing current model stats...")
modelprintStats()

model.write(lp_path)
model.message("Model formulation exported to a .lp file (" + lp_path + ")")
model.write(mps_path)
model.message("Model formulation exported to a .mps file (" + mps_path + ")")

for i in Nodes:
    for j in Nodes:
        if i != j:
            for v in D:
                for s in X:
                    if x_ijvs[i,j,v,s].getattr("UB") == 0:
                        model.remove(x_ijvs[i,j,v,s])

model.update()
if gurobi_optimize:

    model.message("Starting optimization process...")
    model.optimize()

    if model.status == GRB.Status.OPTIMAL:
        if gurobi_printQuality: model.printQuality()
        model.message("\nModel has reached optimality.")
        model.write(sol_path)
        model.message("Optimal solution exported to a .sol file (" + sol_path + ")")

        textfile = open(txtsol_path, 'w')
        textfile.write('# Solution for model ' + model_name + '\n')
        textfile.write('# Objective value = ' + str(round(model.objVal, 5)) + '\n')
        for v in model.getVars():
            if round(v.x, 5) > 0: textfile.write('%s %g\n' % (v.VarName, round(v.x, 5)))
        textfile.close()

    elif model.status == GRB.Status.INFEASIBLE:
        model.message("\nModel is infeasible.")
        if gurobi_genIIS:
            model.computeIIS()

```

```

        model.write(ilp_path)
        model.message("An Irreducible Inconsistent Subsystem (IIS) was written to " + iis_path)
    model.message("\n")
elif model.status == GRB.Status.UNBOUNDED:
    model.message("\nModel is unbounded.\n\n")
elif model.status == GRB.Status.INF_OR_UNBD:
    model.message("\nModel is infeasible or unbounded.")
    if gurobi_genIIS:
        model.computeIIS()
        model.write(ilp_path)
        model.message("An Irreducible Inconsistent Subsystem (IIS) was written to " + iis_path)
elif model.status == GRB.Status.TIME_LIMIT:
    if model.SolCount > 0 and gurobi_printQuality: model.printQuality()
    model.message("\nOptimization terminated because the time expended exceeded the limit.")
    if model.SolCount > 0:
        model.write(sol_path)
        model.message("Incumbent solution was exported to a .sol file (" + sol_path + ")")

    textfile = open(txtsol_path, 'w')
    textfile.write('# Solution for model ' + model_name + '\n')
    textfile.write('# Objective value = ' + str(round(model.objVal, 5)) + '\n')
    for v in model.getVars():
        if round(v.x, 5) > 0: textfile.write('%s %g\n' % (v.VarName, round(v.x, 5)))
    textfile.close()

if gurobi_resetOnEnd: model.reset()

```

Anexo B

Código relativo à criação de instâncias

Neste anexo é apretnado o código *Python* relativo à geração de instâncias, assim como o código em *Excel - Visual Basic*

B.1 Código *Python*

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun  6 11:22:18 2017

@author: Francisco Capa
"""
import json
from openpyxl import *
from math import sin, cos, sqrt, atan2, radians

#fun ao para calcular distancias

def dist_lat_lon(lat1, lat2, lon1, lon2):
    R = 6373.0

    if(lat1 == lat2 and lon1==lon2):
        return 0

    lat1 = radians(lat1)
    lon1 = radians(lon1)
    lat2 = radians(lat2)
    lon2 = radians(lon2)
```

```

dlon = lon2 - lon1
dlat = lat2 - lat1

a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
c = 2 * atan2(sqrt(a), sqrt(1 - a))

return R * c

#open workbook

nomedoficheiro = 'InstanceCreator5.xlsm'

wb = load_workbook(nomedoficheiro , read_only = True)

data = dict()

data['Pile'] = dict()
data['Terminal'] = dict()
data['PowerPlant'] = dict()
data['A-Truck'] = dict()
data['B-Truck'] = dict()
data['Chipper'] = dict()
data['Depot']= dict()
data['Service']= dict()
data['Nodes'] = dict()
data['Timeline'] = dict()

#Timeline
data['Timeline']['startTime'] = 450
data['Timeline']['endTime'] = 10080

#Set of raw material piles at the roadside

#Servi os

```



```

data[ 'Service' ][ 's1' ] = ''
data[ 'Service' ][ 's2' ] = ''
data[ 'Service' ][ 's3' ] = ''
data[ 'Service' ][ 's4' ] = ''
data[ 'Service' ][ 's5' ] = ''
data[ 'Service' ][ 's6' ] = ''
data[ 'Service' ][ 's7' ] = ''
data[ 'Service' ][ 's8' ] = ''
data[ 'Service' ][ 's9' ] = ''
data[ 'Service' ][ 's10' ] = ''
data[ 'Service' ][ 's11' ] = ''

```

#Localizacoes

```
sheet = wb[ 'Locations' ]
```

#Carregar Depot

```
i = 0
```

```
j = 0
```

```
while (sheet.cell(row = i +9, column = j+25).value is not None):
```

```
    data[ 'Depot' ][ sheet.cell(row = i +9, column = j + 25).value ] = dict()
```

```
    data[ 'Depot' ][ sheet.cell(row = i +9, column = j + 25).value ][ 'allservice' ] = 0
```

```
    data[ 'Depot' ][ sheet.cell(row = i +9, column = j + 25).value ][ 'ssimul' ] = 0
```

```
    data[ 'Depot' ][ sheet.cell(row = i +9, column = j + 25).value ][ 'latitude' ] = 0
```

```
    data[ 'Depot' ][ sheet.cell(row = i +9, column = j + 25).value ][ 'longitude' ] = 0
```

```
    i = i+1
```

#CARREGAR PILHAS DO EXCEL

#sheet.cell(row = i, column = j).value

```
i = 0
```

```
j = 0
```

```
while (sheet.cell(row = i +9,column = j + 6).value is not None):
```

```
    if sheet.cell(row = i +9,column = j + 6).value in dict.keys(data[ 'Pile' ]):
```

```
        #data[ 'Pile' ][ sheet.cell(row = i +9, column = j + 6).value ][ 'dummyNodes' ] = 0
```

```
        data[ 'Pile' ][ sheet.cell(row = i +9, column = j + 6).value ][ 'dummyNodes' ] = 0
```

```
        i = i +1
```

```
        continue
```

```
    data[ 'Pile' ][ sheet.cell(row = i +9, column = j + 6).value ] = dict()
```

```
    data[ 'Pile' ][ sheet.cell(row = i +9, column = j + 6).value ][ 'quantityAvailable' ] = 0
```

```
#    data[ 'Pile' ][ sheet.cell(row = i +9, column = j + 6).value ][ 'dummyNodes' ] = 0
```

```

data['Pile'][sheet.cell(row = i + 9, column = j + 6).value]['dummyNodes']=[s
data['Pile'][sheet.cell(row = i + 9, column = j + 6).value]['allservices']=[
data['Pile'][sheet.cell(row = i + 9, column = j + 6).value]['ssimul']=[["s3",
data['Pile'][sheet.cell(row = i + 9, column = j + 6).value]['sprec']=[["s3",
data['Pile'][sheet.cell(row = i + 9, column = j + 6).value]['latitude'] = she
data['Pile'][sheet.cell(row = i + 9, column = j + 6).value]['longitude'] = sh
i = i + 1
#CARREGAR TERMINAIS DO EXCEL
i = 0
j = 0
while (sheet.cell(row = i + 9, column = j + 12).value is not None):
    if sheet.cell(row = i + 9, column = j + 12).value in dict.keys(data['Terminal']):
        data['Terminal'][sheet.cell(row = i + 9, column = j + 12).value]['dummyNodes']=[s
        i = i + 1
        continue
    data['Terminal'][sheet.cell(row = i + 9, column = j + 12).value] = dict()
    data['Terminal'][sheet.cell(row = i + 9, column = j + 12).value]['initialStoc
#data['Terminal'][sheet.cell(row = i + 9, column = j + 12).value]['dummyNodes']=[s
    data['Terminal'][sheet.cell(row = i + 9, column = j + 12).value]['maxCapacity']=[s
    data['Terminal'][sheet.cell(row = i + 9, column = j + 12).value]['dummyNodes']=[s
    data['Terminal'][sheet.cell(row = i + 9, column = j + 12).value]['allservices']=[s
    data['Terminal'][sheet.cell(row = i + 9, column = j + 12).value]['latitude']=[s
    data['Terminal'][sheet.cell(row = i + 9, column = j + 12).value]['longitude']=[s
    i = i + 1

#CARREGAR POWER PLANTS DO EXCEL
i = 0
j = 0
while (sheet.cell(row = i + 9, column = j + 19).value is not None):
    if sheet.cell(row = i + 9, column = j + 19).value in dict.keys(data['PowerPlant']):
        i = i + 1
        continue
    data['PowerPlant'][sheet.cell(row = i + 9, column = j + 19).value] = dict()
    data['PowerPlant'][sheet.cell(row = i + 9, column = j + 19).value]['maxCapacity']=[s
#    data['Pile'][sheet.cell(row = i + 9, column = j + 6).value]['dummyNodes']=[s
    data['PowerPlant'][sheet.cell(row = i + 9, column = j + 19).value]['Demand']=[s
    data['PowerPlant'][sheet.cell(row = i + 9, column = j + 19).value]['allservices']=[s
    data['PowerPlant'][sheet.cell(row = i + 9, column = j + 19).value]['latitude']=[s
    data['PowerPlant'][sheet.cell(row = i + 9, column = j + 19).value]['longitude']=[s
    i = i + 1

```

```
#Ve culos
```

```
sheet = wb[ 'Vehicles' ]
```

```
#Chipper
```

```
i=0
```

```
j=0
```

```
z=0
```

```
while(sheet.cell(row = i + 9,column = j + 5).value is not None):
```

```
    data[ 'Chipper' ][ sheet.cell(row = i + 9,column = j + 5).value ] = dict()
```

```
    data[ 'Chipper' ][ sheet.cell(row = i + 9,column = j + 5).value ][ 'allservice' ] =
```

```
    data[ 'Chipper' ][ sheet.cell(row = i + 9,column = j + 5).value ][ 'sinit' ] = "
```

```
    data[ 'Chipper' ][ sheet.cell(row = i + 9,column = j + 5).value ][ 'sdouble' ] =
```

```
    data[ 'Chipper' ][ sheet.cell(row = i + 9,column = j + 5).value ][ 'productiv' ] =
```

```
    data[ 'Chipper' ][ sheet.cell(row = i + 9,column = j + 5).value ][ 'transport' ] =
```

```
    data[ 'Chipper' ][ sheet.cell(row = i + 9,column = j + 5).value ][ 'fixedCost' ] =
```

```
    z = 0
```

```
    while(sheet.cell(row = 8, column = z +6).value is not None):
```

```
        data[ 'Chipper' ][ sheet.cell(row = i + 9,column = j + 5).value ][ 'p' ] =
```

```
        z = z+1
```

```
    i = i +1
```

```
#ATruck
```

```
colA = z +6 +1
```

```
i = 0
```

```
j = 0
```

```
while(sheet.cell(row = i +9 ,column = colA ).value is not None):
```

```
    data[ 'A-Truck' ][ sheet.cell(row = i+9,column = colA).value ] = dict()
```

```
    data[ 'A-Truck' ][ sheet.cell(row = i+9,column = colA).value ][ 'fixedCost' ] =
```

```
    data[ 'A-Truck' ][ sheet.cell(row = i+9,column = colA).value ][ 'transportCos' ] =
```

```
    data[ 'A-Truck' ][ sheet.cell(row = i+9,column = colA).value ][ 'Capacity' ] =
```

```
    data[ 'A-Truck' ][ sheet.cell(row = i+9,column = colA).value ][ 'allservices' ] =
```

```
    data[ 'A-Truck' ][ sheet.cell(row = i+9,column = colA).value ][ 'sinit' ]= "s7"
```

```
    data[ 'A-Truck' ][ sheet.cell(row = i+9,column = colA).value ][ 'sdouble' ]= [
```

```
    i = i +1
```

```
#BTruck
```

```
colB = colA +4 +1
```

```

i = 0
j = 0
while(sheet.cell(row = i +9, column = colB).value is not None):
    data['B-Truck'][sheet.cell(row = i +9,column = colB).value] = dict()
    data['B-Truck'][sheet.cell(row = i+9,column = colB).value]['fixedCost'] = sh
    data['B-Truck'][sheet.cell(row = i+9,column = colB).value]['transportCost'] =
    data['B-Truck'][sheet.cell(row = i+9,column = colB).value]['Capacity'] = she
    data['B-Truck'][sheet.cell(row = i+9,column = colB).value]['allservices']=["
    data['B-Truck'][sheet.cell(row = i+9,column = colB).value]['sinit']="s4"
    data['B-Truck'][sheet.cell(row = i+9,column = colB).value]['sdouble']=[["pile
    i = i +1

```

#Distancias

```

O = list(data['Depot'].keys())
M = list(data['PowerPlant'].keys())
P = list(data['Pile'].keys())
T = list(data['Terminal'].keys())

```

#From Piles

```

for k in P:
    data['Nodes'][k] = dict()
#to Piles
    for l in P:
        data['Nodes'][k][l] = dict()
        data['Nodes'][k][l]['travelDistance'] = dist_lat_lon(data['Pile'][k]['la
#to Terminal
        for t in T:
            data['Nodes'][k][t] = dict()
            data['Nodes'][k][t]['travelDistance'] = dist_lat_lon(data['Pile'][k]['la
#to Power Plant
            for m in M:
                data['Nodes'][k][m] = dict()
                data['Nodes'][k][m]['travelDistance'] = dist_lat_lon(data['Pile'][k]['la
#to Depot
            for o in O:
                data['Nodes'][k][o] = dict()
                data['Nodes'][k][o]['travelDistance'] = dist_lat_lon(data['Pile'][k]['la

```

```

#From Depot
for o in O:
    data[ 'Nodes' ][o] = dict()
#to Piles
    for l in P:
        data[ 'Nodes' ][o][l] = dict()
        data[ 'Nodes' ][o][l][ 'travelDistance' ] = dist_lat_lon( data[ 'Depot' ][o]
#to Terminal
        for t in T:
            data[ 'Nodes' ][o][t] = dict()
            data[ 'Nodes' ][o][t][ 'travelDistance' ] = dist_lat_lon( data[ 'Depot' ][o]
#to Power Plant
            for m in M:
                data[ 'Nodes' ][o][m] = dict()
                data[ 'Nodes' ][o][m][ 'travelDistance' ] = dist_lat_lon( data[ 'Depot' ][o]
#to Depot
                for p in O:
                    data[ 'Nodes' ][o][p] = dict()
                    data[ 'Nodes' ][o][p][ 'travelDistance' ] = dist_lat_lon( data[ 'Depot' ][o]

#From Terminal
for t in T:
    data[ 'Nodes' ][t] = dict()
#to Piles
    for l in P:
        data[ 'Nodes' ][t][l] = dict()
        data[ 'Nodes' ][t][l][ 'travelDistance' ] = dist_lat_lon( data[ 'Terminal'
#to Terminal
        for z in T:
            data[ 'Nodes' ][t][z] = dict()
            data[ 'Nodes' ][t][z][ 'travelDistance' ] = dist_lat_lon( data[ 'Terminal'
#to Power Plant
            for m in M:
                data[ 'Nodes' ][t][m] = dict()
                data[ 'Nodes' ][t][m][ 'travelDistance' ] = dist_lat_lon( data[ 'Terminal'
#to Depot
                for o in O:
                    data[ 'Nodes' ][t][p] = dict()
                    data[ 'Nodes' ][t][p][ 'travelDistance' ] = dist_lat_lon( data[ 'Terminal'

```

```

#From PowerPlant
for n in M:
    data[ 'Nodes' ][n] = dict()
#to Piles
    for l in P:
        data[ 'Nodes' ][n][l] = dict()
        data[ 'Nodes' ][n][l][ 'travelDistance' ] = dist_lat_lon( data[ 'PowerPlant' ][
#to Terminal
        for t in T:
            data[ 'Nodes' ][n][t] = dict()
            data[ 'Nodes' ][n][t][ 'travelDistance' ] = dist_lat_lon( data[ 'PowerPlant' ][
#to Power Plant
            for m in M:
                data[ 'Nodes' ][n][m] = dict()
                data[ 'Nodes' ][n][m][ 'travelDistance' ] = dist_lat_lon( data[ 'PowerPlant' ][
#to Depot
                for p in O:
                    data[ 'Nodes' ][n][p] = dict()
                    data[ 'Nodes' ][n][p][ 'travelDistance' ] = dist_lat_lon( data[ 'PowerPlant' ][

#dump para JSON

with open( 'teste.json', 'w' ) as fp:
    json.dump( data , fp , sort_keys = False , indent = 3 )
    fp.close

#Fechar excel

wb._archive.close()

```

Anexo C

Soluções alternativas para a instância apresentada

C.1 Solução com menos um veículo tipo A e uma estilhadora

Valor da função objetivo = **14643**

Tabela C.1: Solução com menos um veículo tipo A e uma estilhadora

Veículo	Origem	Destino	Serviço	Execução de s
Tipo A 1	Depósito	Ponto de recolha 2/0	s7	464
Estilhadora 1	Depósito	Ponto de recolha 2/0	s8	464
Tipo B 1	Depósito	Ponto de recolha 2/1	s4	465
Estilhadora 1	Ponto de recolha 2/0	Ponto de recolha 2/1	s3	465
Tipo A 1	Ponto de recolha 2/0	Ponto de recolha 2/1	s1	470.56
Tipo B 1	Ponto de recolha 2/1	Fábrica 1/2	s5	486.56
Tipo A 1	Ponto de recolha 2/1	Ponto de recolha 3/0	s7	494.56
Estilhadora 1	Ponto de recolha 2/1	Ponto de recolha 3/0	s8	494.56
Estilhadora 1	Ponto de recolha 3/0	Ponto de recolha 3/1	s3	495.56
Tipo B 2	Depósito	Ponto de recolha 3/1	s4	495.56
Tipo A 1	Ponto de recolha 3/0	Ponto de recolha 3/1	s1	499.72
Tipo A 1	Ponto de recolha 3/1	Ponto de recolha 1/0	s7	509.72
Estilhadora 1	Ponto de recolha 3/1	Ponto de recolha 1/0	s8	509.72
Tipo B 1	Fábrica 1/2	Ponto de recolha 1/1	s4	513.56
Estilhadora 1	Ponto de recolha 1/0	Ponto de recolha 1/1	s3	513.56
Tipo A 1	Ponto de recolha 1/0	Ponto de recolha 1/1	s1	518.56
Tipo B 2	Ponto de recolha 3/1	Fábrica 2/1	s5	542.72
Tipo A 1	Ponto de recolha 1/1	Depósito	s10	544.56
Estilhadora 1	Ponto de recolha 1/1	Depósito	s11	544.56
Tipo B 1	Ponto de recolha 1/1	Armazém 2/D1	s5	548.56
Tipo B1	Armazém 2/D1	Armazém 2/S1	s9	549.56
Tipo B 2	Fábrica 2/1	Depósito	s6	565.72
Tipo B 1	Armazém 2/S1	Fábrica 1/1	s5	571.56
Tipo B 1	Fábrica 1/1	Armazém 1/S1	s9	626.56
Tipo B 1	Armazém 1/S1	Fábrica 3/1	s5	703.56
Tipo B 1	Fábrica 3/1	Depósito	s6	733.56

C.2 Solução com menos um veículo do tipo B

Valor da função objetivo = **16411.3333**

Tabela C.2: Solução com menos um veículo do tipo B

Veículo	Origem	Destino	Serviço	Execução de s
Tipo A 2	Depósito	Ponto de recolha 1/0	s7	475
Estilhadora 2	Depósito	Ponto de recolha 1/0	s8	475
Tipo B 1	Depósito	Ponto de recolha 1/1	s4	476
Estilhadora 2	Ponto de recolha 1/0	Ponto de recolha 1/1	s3	476
Tipo A 2	Ponto de recolha 1/0	Ponto de recolha 1/1	s1	510
Tipo B 1	Ponto de recolha 1/1	Armazém 2/D1	s5	511
Tipo A 1	Depósito	Ponto de recolha 2/0	s7	529
Estilhadora 1	Depósito	Ponto de recolha 2/0	s8	529
Estilhadora 1	Ponto de recolha 2/0	Ponto de recolha 2/1	s3	530
Tipo B 1	Armazém 2/D1	Ponto de recolha 2/1	s4	530
Tipo A 1	Ponto de recolha 2/0	Ponto de recolha 2/1	s1	535.56
Tipo B 1	Ponto de recolha 2/1	Fábrica 1/2	s5	551.56
Estilhadora 2	Ponto de recolha 1/1	Ponto de recolha 3/0	s8	583.56
Tipo A 2	Ponto de recolha 1/1	Ponto de recolha 3/0	s7	583.56
Tipo B 1	Fábrica 2/1	Ponto de recolha 3/1	s4	584.56
Estilhadora 2	Ponto de recolha 3/0	Ponto de recolha 3/1	s3	584.56
Tipo A 2	Ponto de recolha 3/0	Ponto de recolha 3/1	s1	588.72
Estilhadora 2	Ponto de recolha 3/1	Depósito	s11	617.72
Tipo A 1	Ponto de recolha 2/1	Depósito	s10	617.72
Estilhadora 1	Ponto de recolha 2/1	Depósito	s11	617.72
Tipo A 2	Ponto de recolha 3/1	Depósito	s10	617.72
Tipo B 1	Ponto de recolha 3/1	Fábrica 2/1	s5	631.72
Tipo B 1	Fábrica 2/1	Armazém 2/S1	s9	654.72
Tipo B 1	Armazém 2/S1	Fábrica 1/1	s5	676.72
Tipo B 1	Fábrica 1/1	Armazém 1/S1	s9	731.72
Tipo B 1	Armazém 1/S1	Fábrica 3/1	s5	808.72
Tipo B 1	Fábrica 3/1	Depósito	s6	838.72

C.3 Solução com menos um veículo de cada tipo

Valor da função objetivo = **10216.11**

Tabela C.3: Solução com menos um veículo de cada tipo

Veículo	Origem	Destino	Serviço	Execução de s
Tipo A 1	Depósito	Ponto de recolha 2/0	s7	464
Estilhadora 1	Depósito	Ponto de recolha 2/0	s8	464
Tipo B 1	Depósito	Ponto de recolha 2/1	s4	465
Estilhadora 1	Ponto de recolha 2/0	Ponto de recolha 2/1	s3	465
Tipo A 1	Ponto de recolha 2/0	Ponto de recolha 2/1	s1	470.56
Tipo B 1	Ponto de recolha 2/1	Fábrica 1/2	s5	486.56
Tipo A 1	Ponto de recolha 2/1	Ponto de recolha 1/0	s7	488.56
Estilhadora 1	Ponto de recolha 2/1	Ponto de recolha 1/0	s8	488.56
Estilhadora 1	Ponto de recolha 1/0	Ponto de recolha 1/1	s3	513.56
Tipo B 1	Fábrica 1/2	Ponto de recolha 1/1	s4	513.56
Tipo B 1	Ponto de recolha 1/1	Armazém 2/D1	s5	548.56
Tipo A 1	Ponto de recolha 1/0	Ponto de recolha 1/1	s1	574.56
Estilhadora 1	Ponto de recolha 1/1	Ponto de recolha 3/0	s8	583.56
Tipo A 1	Ponto de recolha 1/1	Ponto de recolha 3/0	s7	583.56
Tipo B 1	Armazém 1/D1	Ponto de recolha 3/1	s4	584.56
Estilhadora 1	Ponto de recolha 3/0	Ponto de recolha 3/1	s3	584.56
Tipo A 1	Ponto de recolha 3/0	Ponto de recolha 3/1	s1	588.72
Tipo A 1	Ponto de recolha 3/1	Depósito	s10	618.72
Estilhadora 1	Ponto de recolha 3/1	Depósito	s11	618.72
Tipo B 1	Ponto de recolha 3/1	Fábrica 2/1	s5	631.72
Tipo B 1	Fábrica 2/1	Armazém 2/S1	s9	654.72
Tipo B 1	Armazém 2/S1	Fábrica 1/1	s5	676.72
Tipo B 1	Fábrica 1/1	Armazém 1/S1	s9	731.72
Tipo B 1	Armazém 1/S1	Fábrica 3/1	s5	808.72
Tipo B 1	Fábrica 3/1	Depósito	s6	838.72

Referências

- [1] A.A. Rentizelas, I.P. Tatsiopoulos, e A. Tolis. An optimization model for multi-biomass tri-generation energy supply. *Biomass and Bioenergy*, 33(2):223–233, Fevereiro 2009. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0961953408001463>, doi:10.1016/j.biombioe.2008.05.008.
- [2] Fereshteh Mafakheri e Fuzhan Nasiri. Modeling of biomass-to-energy supply chain operations: Applications, challenges and research directions. *Energy Policy*, 67:116–126, Abril 2014. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0301421513012019>, doi:10.1016/j.enpol.2013.11.071.
- [3] Karin Ericsson e Sven Werner. The introduction and expansion of biomass use in Swedish district heating systems. *Biomass and Bioenergy*, 94:57–65, Novembro 2016. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0961953416302793>, doi:10.1016/j.biombioe.2016.08.011.
- [4] P Flisberg, M Frisk, e M Rönnqvist. FuelOpt: a decision support system for forest fuel logistics. *Journal of the Operational Research Society*, 63(11):1600–1612, Novembro 2012. URL: <http://link.springer.com/10.1057/jors.2011.157>, doi:10.1057/jors.2011.157.
- [5] Annelies De Meyer, Joana Almeida, Wouter Achten, Bart Muys, Dirk Cattrysse, e Jos Van Orshoven. Incorporating life cycle impact assessment in a mathematical model to optimize strategic decisions in biomass-for-bioenergy supply chains. Em *Fulfilling LCA's Promise, Proceedings from the LCA XIII International Conference*, páginas 24–33. American Center for Life Cycle Assessment, 2013. URL: <https://lirias.kuleuven.be/handle/123456789/420668>.
- [6] Heungjo An, Wilbert E. Wilhelm, e Stephen W. Searcy. Biofuel and petroleum-based fuel supply chain research: A literature review. *Biomass and Bioenergy*, Julho 2011. URL: <http://linkinghub.elsevier.com/retrieve/pii/S096195341100345X>, doi:10.1016/j.biombioe.2011.06.021.
- [7] Helene Gunnarsson, Mikael Rönnqvist, e Jan T Lundgren. Supply chain modelling of forest fuel. *European Journal of Operational Research*, 158(1):103–123, Outubro 2004. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0377221703003540>, doi:10.1016/S0377-2217(03)00354-0.
- [8] George B. Dantzig e John H. Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959. URL: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.6.1.80>.

- [9] Bala Chandran e S. Raghavan. Modeling and solving the capacitated vehicle routing problem on trees. Em *The Vehicle Routing problem: Latest advances and new challenges*, páginas 239–261. Springer, 2008. URL: http://link.springer.com/chapter/10.1007/978-0-387-77778-8_11.
- [10] Javier Faulin, Angel Juan, Fernando Lera, e Scott Grasman. Solving the Capacitated Vehicle Routing Problem with Environmental Criteria Based on Real Estimations in Road Transportation: A Case Study. *Procedia - Social and Behavioral Sciences*, 20:323–334, 2011. URL: <http://linkinghub.elsevier.com/retrieve/pii/S1877042811014182>, doi:10.1016/j.sbspro.2011.08.038.
- [11] Alvina G.H. Kek, Ruey Long Cheu, e Qiang Meng. Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Mathematical and Computer Modelling*, 47(1-2):140–152, Janeiro 2008. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0895717707001215>, doi:10.1016/j.mcm.2007.02.007.
- [12] Tonci Caric e Hrvoje Gold. *Vehicle routing problem*. InTech, Rijek, Croatia, 2008. OCLC: 682082977. URL: http://www.intechweb.org/books/show/title/vehicle_routing_problem.
- [13] Paolo Toth e Daniele Vigo, editores. *Vehicle routing: problems, methods, and applications*. Número 18 em MOS-SIAM series on optimization. SIAM [u.a.], Philadelphia, Pa, 2. ed edição, 2014. OCLC: 935554111.
- [14] Nasser A. El-Sherbeny. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University - Science*, 22(3):123–131, Julho 2010. URL: <http://linkinghub.elsevier.com/retrieve/pii/S1018364710000297>, doi:10.1016/j.jksus.2010.03.002.
- [15] Jacques Desrosiers, François Soumis, e Martin Desrochers. Routing with time windows by column generation. *Networks*, 14(4):545–565, 1984. URL: <http://onlinelibrary.wiley.com/doi/10.1002/net.3230140406/full>.
- [16] Martin Desrochers, Jacques Desrosiers, e Marius M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342–354, Março 1992. doi:10.1287/opre.40.2.342.
- [17] Eunjeong Choi e Dong-Wan Tcha. A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34(7):2080–2095, Julho 2007. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0305054805002650>, doi:10.1016/j.cor.2005.08.002.
- [18] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, e Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):pp. 316–329, 1998. URL: <http://www.jstor.org/stable/222825>.
- [19] Alberto V. Donati, Roberto Montemanni, Luca M. Gambardella, e Andrea E. Rizzoli. Integration of a robust shortest path algorithm with a time dependent vehicle routing model and applications. Em *Computational Intelligence for Measurement Systems and Applications, 2003. CIMSA'03. 2003 IEEE International Symposium on*, páginas 26–31. IEEE, 2003. URL: <http://ieeexplore.ieee.org/abstract/document/1227196/>.

- [20] Tuncay Yigit e Ozkan Unsal. Using the Ant Colony Algorithm for Real-Time Automatic Route of School Buses. *International Arab Journal of Information Technology (IAJIT)*, 13(5), 2016. URL: <http://ccis2k.org/iajit/PDF/vol.13,%20no.5/8072.pdf>.
- [21] Fred Glover. Tabu Search—Part I. *ORSA Journal on Computing*, 1(3):190–206, Agosto 1989. URL: <http://pubsonline.informs.org/doi/abs/10.1287/ijoc.1.3.190>, doi:10.1287/ijoc.1.3.190.
- [22] Fred Glover. Tabu Search—Part II. *ORSA Journal on Computing*, 2(1):4–32, Fevereiro 1990. URL: <http://pubsonline.informs.org/doi/abs/10.1287/ijoc.2.1.4>, doi:10.1287/ijoc.2.1.4.
- [23] Christophe Duhamel, Jean-Yves Potvin, e Jean-Marc Rousseau. A Tabu Search Heuristic for the Vehicle Routing Problem with Backhauls and Time Windows. *Transportation Science*, 31(1):49–59, Fevereiro 1997. URL: <http://pubsonline.informs.org/doi/abs/10.1287/trsc.31.1.49>, doi:10.1287/trsc.31.1.49.
- [24] J. Pirie Hart e Andrew W. Shogan. Semi-greedy heuristics: An empirical study. *Operations Research Letters*, 6(3):107–114, 1987. URL: <http://www.sciencedirect.com/science/article/pii/0167637787900216>.
- [25] Thomas A. Feo e Mauricio GC Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8(2):67–71, 1989. URL: <http://www.sciencedirect.com/science/article/pii/0167637789900023>.
- [26] Vera C. Hemmelmayr, Jean-François Cordeau, e Teodor Gabriel Crainic. An adaptive large neighborhood search heuristic for Two-Echelon Vehicle Routing Problems arising in city logistics. *Computers & Operations Research*, 39(12):3215–3228, Dezembro 2012. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0305054812000871>, doi:10.1016/j.cor.2012.04.007.
- [27] M. Drexl. *On Some Generalized Routing Problems*. 2007. URL: <https://books.google.pt/books?id=M48vOAAACAAJ>.
- [28] Michael Drexl. Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments. *Networks*, 63(1):119–133, Janeiro 2014. URL: <http://doi.wiley.com/10.1002/net.21526>, doi:10.1002/net.21526.
- [29] Marjolein Veenstra, Marilène Cherkesly, Guy Desaulniers, e Gilbert Laporte. The pickup and delivery problem with time windows and handling operations. *Computers & Operations Research*, 77:127–140, Janeiro 2017. URL: <http://linkinghub.elsevier.com/retrieve/pii/S030505481630185X>, doi:10.1016/j.cor.2016.07.014.
- [30] Gerardo Berbeglia, Jean-François Cordeau, Irina Gribkovskaia, e Gilbert Laporte. Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15(1):1–31, Maio 2007. URL: <http://link.springer.com/10.1007/s11750-007-0009-0>, doi:10.1007/s11750-007-0009-0.
- [31] Nizar El Hachemi, Issmail El Hallaoui, Michel Gendreau, e Louis-Martin Rousseau. Flow-based integer linear programs to solve the weekly log-truck scheduling problem. *Annals of Operations Research*, Janeiro 2014. URL: <http://link.springer.com/10.1007/s10479-014-1527-4>, doi:10.1007/s10479-014-1527-4.

- [32] Greg Rix, Louis-Martin Rousseau, e Gilles Pesant. Solving a multi-period log-truck scheduling problem with column generation. Em *34th Council on Forest Engineering Annual Meeting. Quebec, QC, Canada*, 2011. URL: <https://www.cirrelet.ca/COFE2011/proceedings/29-rix.pdf>.
- [33] Moshe Dror e Pierre Trudeau. Savings by Split Delivery Routing. *Transportation Science*, 23(2):141–145, Maio 1989. URL: <http://pubsonline.informs.org/doi/abs/10.1287/trsc.23.2.141>, doi:10.1287/trsc.23.2.141.
- [34] Sung Hun Song, Kwan Suk Lee, e Gi Sub Kim. A practical approach to solving a newspaper logistics problem using a digital map. *Computers & industrial engineering*, 43(1):315–330, 2002. URL: <http://www.sciencedirect.com/science/article/pii/S0360835202000773>.
- [35] Si Chen, Bruce Golden, e Edward Wasil. The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks*, 49(4):318–329, Julho 2007. URL: <http://doi.wiley.com/10.1002/net.20181>, doi:10.1002/net.20181.
- [36] C. Archetti, M. G. Speranza, e A. Hertz. A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem. *Transportation Science*, 40(1):64–73, Fevereiro 2006. URL: <http://pubsonline.informs.org/doi/abs/10.1287/trsc.1040.0103>, doi:10.1287/trsc.1040.0103.
- [37] Zizhen Zhang, Huang He, Zhixing Luo, Hu Qin, e Songshan Guo. An efficient forest-based tabu search algorithm for the split-delivery vehicle routing problem. Em *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15*, páginas 3432–3438. AAAI Press, 2015. URL: <http://dl.acm.org/citation.cfm?id=2888116.2888193>.
- [38] Michael Drexl. Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints. *Transportation Science*, 46(3):297–316, Agosto 2012. URL: <http://pubsonline.informs.org/doi/abs/10.1287/trsc.1110.0400>, doi:10.1287/trsc.1110.0400.
- [39] Jesus Gonzalez-Feliu, Guido Perboli, Roberto Tadei, e Daniele Vigo. The two-echelon capacitated vehicle routing problem. 2008. URL: <https://halshs.archives-ouvertes.fr/halshs-00879447/>.
- [40] Cristián E. Cortés, Martín Matamala, e Claudio Contardo. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3):711–724, Fevereiro 2010. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0377221709000356>, doi:10.1016/j.ejor.2009.01.022.
- [41] Min Wen, Jesper Larsen, Jens Clausen, Jean-François Cordeau, e Gilbert Laporte. Vehicle routing with cross-docking. *Journal of the Operational Research Society*, 60(12):1708–1718, 2009. URL: <http://link.springer.com/article/10.1057/jors.2008.108>.
- [42] Stefan Bock. Real-time control of freight forwarder transportation networks by integrating multimodal transport chains. *European Journal of Operational Research*, 200(3):733–746, Fevereiro 2010. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0377221709000411>, doi:10.1016/j.ejor.2009.01.046.

- [43] Tore Grünert e Hans-Jürgen Sebastian. Planning models for long-haul operations of postal and express shipment companies. *European Journal of Operational Research*, 122(2):289–309, 2000. URL: <http://www.sciencedirect.com/science/article/pii/S0377221799002349>.
- [44] Christopher Mues e Stefan Pickl. Transshipment and time windows in vehicle routing. Em *Parallel Architectures, Algorithms and Networks, 2005. ISPAN 2005. Proceedings. 8th International Symposium on*, páginas 6–pp. IEEE, 2005. URL: <http://ieeexplore.ieee.org/abstract/document/1575814/>.
- [45] Teodor Gabriel Crainic, Nicoletta Ricciardi, e Giovanni Storch. Models for Evaluating and Planning City Logistics Systems. *Transportation Science*, 43(4):432–454, Novembro 2009. URL: <http://pubsonline.informs.org/doi/abs/10.1287/trsc.1090.0279>, doi:10.1287/trsc.1090.0279.
- [46] Majid Aldaihani e Maged M. Dessouky. Hybrid scheduling methods for paratransit operations. *Computers & Industrial Engineering*, 45(1):75–96, Junho 2003. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0360835203000329>, doi:10.1016/S0360-8352(03)00032-9.
- [47] Snežana Mitrović-Minić e Gilbert Laporte. The Pickup And Delivery Problem With Time Windows And Transshipment. *INFOR: Information Systems and Operational Research*, 44(3):217–227, Agosto 2006. URL: <http://www.tandfonline.com/doi/full/10.1080/03155986.2006.11732749>, doi:10.1080/03155986.2006.11732749.
- [48] Anders Dohn, Esben Kolind, e Jens Clausen. The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. *Computers & Operations Research*, 36(4):1145–1157, Abril 2009. URL: <http://linkinghub.elsevier.com/retrieve/pii/S030505480700278X>, doi:10.1016/j.cor.2007.12.011.
- [49] Hong K. Chung e John P. Norback. A Clustering and Insertion Heuristic Applied to a Large Routeing Problem in Food Distribution. *Journal of the Operational Research Society*, 42(7):555–564, 1991. URL: <http://dx.doi.org/10.1057/jors.1991.114>, doi:10.1057/jors.1991.114.
- [50] Benoît Laurent e Jin-Kao Hao. Simultaneous vehicle and driver scheduling: A case study in a limousine rental company. *Computers & Industrial Engineering*, 53(3):542–558, Outubro 2007. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0360835207000897>, doi:10.1016/j.cie.2007.05.011.
- [51] Byung-In Kim, Jeongin Koo, e Junhyuk Park. The combined manpower-vehicle routing problem for multi-staged services. *Expert Systems with Applications*, 37(12):8424–8431, Dezembro 2010. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0957417410004471>, doi:10.1016/j.eswa.2010.05.036.
- [52] B.L. Hollis, M.A. Forbes, e B.E. Douglas. Vehicle routing and crew scheduling for metropolitan mail distribution at Australia Post. *European Journal of Operational Research*, 173(1):133–150, Agosto 2006. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0377221705000329>, doi:10.1016/j.ejor.2005.01.005.
- [53] Raymond K. Cheung, Ning Shi, Warren B. Powell, e Hugo P. Simao. An attribute–decision model for cross-border drayage problem. *Transportation Research*

- Part E: Logistics and Transportation Review*, 44(2):217–234, Março 2008. URL: <http://linkinghub.elsevier.com/retrieve/pii/S1366554507000774>, doi:10.1016/j.tre.2007.07.009.
- [54] Christoph Hemsch e Stefan Irnich. Vehicle Routing Problems with Inter-Tour Resource Constraints. Em Bruce Golden, S. Raghavan, e Edward Wasil, editores, *The Vehicle Routing Problem: Latest Advances and New Challenges*, páginas 421–444. Springer US, Boston, MA, 2008. DOI: 10.1007/978-0-387-77778-8_19. URL: http://dx.doi.org/10.1007/978-0-387-77778-8_19.
- [55] Nizar El Hachemi, Michel Gendreau, e Louis-Martin Rousseau. A hybrid constraint programming approach to the log-truck scheduling problem. *Annals of Operations Research*, 184(1):163–178, Abril 2011. URL: <http://link.springer.com/10.1007/s10479-010-0698-x>, doi:10.1007/s10479-010-0698-x.
- [56] Rosanna Tarricone, Agis D. Tsouros, World Health Organization, e Università commerciale Luigi Bocconi, editores. *Home care in Europe: the solid facts*. World Health Organization, Europe, Copenhagen, 2008. OCLC: ocn299702993.
- [57] Chananek Akjiratikarl, Pisal Yenradee, e Paul R. Drake. PSO-based algorithm for home care worker scheduling in the UK. *Computers & Industrial Engineering*, 53(4):559–583, Novembro 2007. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0360835207000927>, doi:10.1016/j.cie.2007.06.002.
- [58] Matias Sevel Rasmussen, Tor Justesen, Anders Dohn, e Jesper Larsen. The Home Care Crew Scheduling Problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3):598–610, Junho 2012. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0377221711009891>, doi:10.1016/j.ejor.2011.10.048.
- [59] Andrea Trautsumwieser e Patrick Hirsch. Optimization of daily scheduling for home health care services. *Journal of Applied Operational Research*, 3(3):124–136, 2011. URL: <http://www.orlabanalytics.ca/jaor/archive/v3/n3/jaorv3n3p124.pdf>.
- [60] Yannick Kergosien, Christophe Lenté, e Jean-Charles Billaut. Home health care problem: An extended multiple traveling salesman problem. Em *4th multidisciplinary international conference on scheduling: Theory and applications (MISTA'09), Dublin (Irlande)*, páginas 10–12, 2009. URL: <http://www.asap.cs.nott.ac.uk/external/mista/2009/papers/085-092-110-P.pdf>.
- [61] Dorota Slawa Mankowska, Frank Meisel, e Christian Bierwirth. The home health care routing and scheduling problem with interdependent services. *Health Care Management Science*, 17(1):15–30, Março 2014. URL: <http://link.springer.com/10.1007/s10729-013-9243-1>, doi:10.1007/s10729-013-9243-1.
- [62] Ricardo Soares. Working paper.